Neural Approaches to Historical Word Reconstruction

Clémentine Fourrier

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Inria

Neural Approaches to Historical Word Reconstruction
(Looking at words through space and time)

Soutenue par
Clémentine Fourrier
Le 26 septembre 2022

École doctorale n°472
École Pratique des Hautes Études

Spécialité
Informatique

Composition du jury :

Sylvain KAHANE Professeur des Universités, Modyco, Université Paris Nanterre & CNRS	*Président*
Marie CANDITO Maîtresse de Conférences, Université Paris Cité	*Rapportrice*
Johann-Mattis LIST Senior Scientist, Max Planck Institute	*Rapporteur*
Julia IVE Lecturer, Queen Mary University of London	*Examinatrice*
Yves SCHERRER Lecturer, University of Helsinki	*Examinateur*
Benoît SAGOT Directeur de Recherche, Inria	*Directeur*
Laurent ROMARY Directeur de Recherche, Inria	*Directeur*
Rachel BAWDEN Chargée de Recherche, Inria	*Encadrante*

École Pratique
des Hautes Études

PSL

Contents

List of Figures

List of Tables

List of Equations

List of Acronyms

BiRNN	Bidirectionnal Recurrent Neural Network
BLEU	BiLingual Evaluation Understudy
BPE	byte-pair encoding
FFNN	Feed Forward Neural Network
GRU	Gated Recurrent Unit
IPA	International Phonetic Alphabet
LSTM	Long Short Term Memory unit
MLP	Multi Layer Perceptron
MT	machine translation
NMT	neural machine translation
POS	Part-of-Speech
RNN	Recurrent Neural Network
ROUGE	Recall Oriented Understudy for Gisting Evaluation
SMT	statistical machine translation
SRNN	Simple Recurrent Neural Network

Notations

Operators

Symbol	Name	Meaning
A $\odot$ B	Hadamard product (Element wise matrix multiplication)	$(A \odot B)_{ij} = A_{ij} B_{ij}$
AB	Matrix multiplication	$(AB)_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}$
A + B	Element wise matrix sum	$(A+B)_{ij} = A_{ij} + B_{ij}$
$\#L$	Number of elements in the vector L (if L is a layer, number of neurons)	
A^T	Transposed of matrix A	$A_{ij}^T = A_{ji}$

Matrices and vectors

Symbol	Meaning
h_t	recurrent unit hidden memory state value at update t (when looking at the input sequence's t^{th} element)
net_i^l	output of the i^{th} neuron of the l^{th} layer
W	weight matrix
W_G	weight matrix of gate G
$W_{ij}^{(l)}$	weight matrix connecting the j^{th} input to the l^{th} layer's i^{th} neuron
v	weight vector
X	input sequence of m elements
x_i	i^{th} element of sequence X
Y	target sequence of n elements
y_i	i^{th} element of sequence Y
$\hat{Y}$	predicted output sequence of $\hat{n}$ elements
$\hat{y}_i$	i^{th} element of sequence $\hat{Y}$

Reference functions

Symbol	Name	Meaning
MSE	Mean squared error	$\text{MSE} = \sum_{i=1}^{\#X} \frac{1}{2}(\hat{Y}_i - Y_i^2)$, for an input X, a prediction $\hat{Y}$ and a target Y
$\sum$	Sum operator	$\sum_{i=0}^{n} x_i = x_0 + x_1 + \ldots + x_i + \ldots + x_n$
σ	Softmax	$\sigma_j(x_j) = \frac{exp(x_j)}{\sum_{k=1}^{N} exp(x_k)}$
S	Sigmoid	A sigmoid function (such as a logistic function, hyperbolic tangent, ...)
tanh	Hyperbolic tangent	$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Acknowledgements

> "But I don't want to go among mad people," Alice remarked.
> "Oh, you can't help that," said the Cat: "we're all mad here. I'm mad. You're mad."
> "How do you know I'm mad?" said Alice.
> "You must be," said the Cat, "or you wouldn't have come here."
>
> Caroll (1865)

As I am writing this final (for me) and first (for you) chapter, the forest outside my window is glowing in the evening sun, and all my books have been unpacked. At last, I am settled.

Doing a PhD has been a fun, interesting, and infuriating experience, and I loved working on my topic, which you will discover in the following pages if you want. (Spoilers: it's looking at how the sounds of words change through time, and the best neural networks to fit this task.) I would like to deeply thank the members of my committee, who provided me with insightful comments and patient questions, helping me make this manuscript better and clearer than it was.

This overall adventure would not have been possible without my thesis supervisors, Benoît, Rachel and Laurent, whom I thank for all their time, expertise, moral support, plus all they taught me on the side during these years. I would also like to thank Douwe, who supervised my last year internship at HuggingFace with deep care and enthusiasm – this internship was a change of environment and pace I deeply needed. Benoît and Douwe both have my gratitude for betting on my profile, as I am, by trade a software engineer, by training a geologist, and now officially a machine learning researcher!

Special thoughts go to the members of my successive teams at Inria: Igor, Ninon and Simona at ARAMIS, my first professional introduction to academia; Cédric, Mauricio, Nathalie, Pierre-Guillaume, Sébastien, Simon and Thierry in the SED, always sharing stories and advice; last but not least, the generations of pioupiouxes turning dinosaurs at ALMAnaCH, notably Alix, Arij, Hugo, Loïc, Louis, Mathilde, Pedro, Syrielle, Tanti and Yoann. You contributed to making these years even more enjoyable. (As did The Caucus, héhé.) Thanks also to my colleagues at HuggingFace, for their warm welcome and trust. I'm looking forward to keeping working with you all!

Puisqu'on ne fait pas des listes de remerciements tous les jours, je voudrais aussi remercier tous les enseignants en sciences, français, et les documentalistes que j'ai rencontré.e.s lors de ma scolarité (en particulier en CM2 et 6e, des années determinantes pour l'identité scolaire, puis

durant tout mon lycée, des années determinantes pour l'orientation); vous avez contribué à me faire aimer ces sujets et avez encouragé mon envie de questions.

Merci à Gabrielle, Marjorie, et Morgane, pour toutes les conversations qu'on a eues et pour tout ce qu'on a vécu. J'espère que nos amitiés dureront encore au moins ce qu'elles ont déjà tenu. Merci également à ma famille proche pour leur soutien inconditionnel et pour ce qui m'a été transmis par chacun: l'amour des livres, le goût de la vulgarisation, une appréciation pour le bidouillage informatique, ainsi que l'humour, l'empathie et la volonté de faire ce qui doit être fait. Cette thèse n'aurait aussi pas été la même sans des suggestions de mon père sur la création de mes données artificielles en première année, des conversations avec ma mère à la fin de ma deuxième année, et une relecture attentive du manuscrit par ma soeur il y a quelques mois. Un merci tout particulier à Pem pour ton soutien moral, nos discussions, la good food, et tout ce qu'on s'apporte au quotidien. The best is yet to come.

Enfin, je dédie cette thèse à Marine: "Pourquoi pas? Ca avait l'air fun!" (avec du savon).

INTRODUCTION

> Montag picked a single small volume from the floor. "Where do we begin?" He opened the book halfway and peered at it. "We begin by beginning, I guess."
>
> Bradbury (1953)

Most readers will have an idea of what etymology is, that some words are 'genetically' related while languages are seemingly ordered in families. But how did we come to know that? Coeurdoux (1768),[1] by rigorously comparing Sanskrit (which he called 'Sanskroustan'), Latin and Greek words, underlined phonetic regularities between these languages — he then discussed these similarities, feeling they could come from either exchange of words caused by commerce, science, geographic proximity, religion or invasion, or, on the other hand, from a common origin to all languages, the latter option having his preference.[2] Some years later, Jones (1786) expressed a similar sentiment, comparing Greek, Latin and Sanskrit. Despite at least many decades of prior work (see (Blench 2004) for an overview), this moment is usually seen as the beginning of the field of ***comparative linguistics***, which, as its name indicates, compares languages, in order to establish their relations. It took another hundred years, and the works of the Neogrammarians (Osthoff and Brugmann 1878), to formalise its main empirical hypothesis, the ***regularity of sound changes***.

This hypothesis states that the lexicon of a language evolves diachronically according to regular and exceptionless sound changes, notwithstanding lexical creation and borrowing mechanisms, which, in laymans terms, gives: the pronunciation of words of a given language changes through time, and if a given sound changes a certain way in some words, it will change the same way in all other similarly-sounding words that contain it (for the given language) — new words, created from scratch or taken from other languages, will follow the changes occuring in their language after they appear. For example, the sequence [ka] in Vulgar Latin changed into [ʧa] in Old French, then to [ʃa] in French. This is illustrated by ***chat*** [ʃa] 'cat' < ***cattus*** *[kat.tʊs] and ***blanche*** [blɑ̃ʃ] 'white (fem.)' < ***blanca*** *[blan.ka]. Such a change is called a ***sound law***. The phonetic history of a language from an earlier to a later stage can then be modelled as an ordered sequence of sound laws.

These sound laws are usually identified by looking at specific word sets, called ***cognates***: given two languages with a common ancestor, two words are said to be cognates if they are an evolution of the same word from said ancestor, called their ***proto-form***. To look at our former example, French ***chat*** 'cat' is cognate with Spanish ***gatto*** 'id.', as they both descend from Latin ***cattus***, 'id.', their common ancestor. The phonological differences between two cognates and their

[1] He belonged to a long line of missionaries commenting on similarities between Indian languages, Greek, and Latin from the XVIth century (Auroux et al. 2008)

[2] He poetically writes that, from a common origin, "Plusieurs termes communs restèrent dans les langues nouvelles; un grand nombre se sont perdus par le laps du temps; d'autres ont été défigurés à un point qu'ils ne sont plus reconnoissables. Quelques-uns ont échappé à ce naufrage, pour être aux hommes un mémorial éternuel de leur commune origine et de leur antique fraternité.", which I will translate as "Many common terms remained in the new languages; a great number of those are now lost to time; when others have been disfigured to a point of non recognition. But some have evaded this shipwreck, to provide men with an eternal memorial of their common origins and antique fraternity.".

languages can be modelled as a group of ***sound correspondences***, and they capture some of the differences between the phonetic evolution of the languages.

However, like Coeurdoux and Jones in their time, we only have access to languages that are either spoken nowadays or have been well conserved (such as Sanskrit, Ancient Greek and Latin, to name a few), so how do we know which words are cognates and that they share a common ancestor? From sound patterns between languages, plausibly related words are identified, and in some cases, their common ancestors are found or reconstructed, making the original words cognates — but all new cognates must obey all the previously identified sound rules, and a situation which presents an exception to a given rule must amend the rule, and if not possible, either invalidate it or more likely consider the proposed cognates as incorrect. From correct cognates, new sound correspondences are identified, from which new cognates will be identified further, and so forth. It is, in a sense, a 'chicken or egg' problem, though the exceptionless nature of sound change constrains the problem enough to make it possible to solve.

All etymological connections we now know derive from the linguistic relations identified through the comparative method (Durkin 2015). This makes cognates very important words to better understand language as a whole. Cognates have also been used in more practical ways, to improve translation systems for languages with little resources (Grönroos et al. 2018; Mann and Yarowsky 2001), or to help field linguists look for plausible words in endangered languages they study (Bodt et al. 2018; Bodt and List 2019).

The two main tasks of the comparative method are therefore cognate identification (finding which words are cognates in a multilingual set of words) and cognate prediction (producing a likely shape for unseen cognate words or their parents): they constitute the basis from which sound correspondences can be found, and from said correspondences, language relations can be inferred. They also are closely related: cognate identification produces cognates sets, which can be used to train cognate prediction, and the possible predicted cognates can then be looked for in data, to help cognate identification. (Chicken or egg, again).

Since the creation of the field, cognates have been studied 'manually' by expert linguists, with in-depth knowledge of several related languages. However, in the last decades, they also benefited from the advances of quantitative or computational techniques, from the prior works of Swadesh (1950) and Swadesh (1955) introducing the first versions of glottochronology and lexicostatistics, albeit in a currently outdated form, trying to date language relationships through their vocabulary stability and evolution, to more recently combinations of statistics, alignement tools, and sometimes neural networks, to specifically identify or predict cognates (Ciobanu and Dinu 2020; Dekker and Zuidema 2021; Frunza and Inkpen 2009; Hauer and Kondrak 2011; List et al. 2017; Mitkov et al. 2007; Mulloni 2007; Rama 2016).

Neural networks are tools that are very good at latently identifying useful underlying patterns when learning on data, for example spontaneously tracking word boundaries when trained on language modelling on unsegmented text (Hahn and Baroni 2019), or more generally implicitly capturing surface, syntactic and semantic knowledge when trained on unrelated tasks such as machine translation or language modelling (Conneau et al. 2018; Jo and Myaeng 2020; Raganato and Tiedemann 2018), to name a few. Therefore, the intuition behind this work was that neural networks, trained on cognate prediction, therefore learning sound correspondences from languages to other languages, were likely to be able to latently learn interesting information for historical linguistics. We were particularly interested in seeing what encoder-decoder

models, standard in machine translation, could produce on the task. However, at the beginning of this work, in 2019, unpublished preliminary experiments as well as other works (Dekker 2018) seemed to conclude that using complex neural networks was likely not a good direction for cognate prediction, as they seemed outperformed by statistical models in our case or simplistic neural networks (perceptrons) in other works, though at the same time, these models of interest were also being used successfully for proto-form reconstruction (Meloni et al. 2021) (on arxiv in 2019). As we believed in the potential of those methods, we decided to investigate the situation more in depth, first by understanding if using complex neural networks for cognate prediction was possible or not, then using it on historical data, before trying to understand the latent information plausibly learnt.

Our core results are showing, first that machine translation inspired neural networks can be successfully used for the task of cognate prediction, when at least 500 training word pairs are available and as long as optimisation is done on the parameters; then, that such models in fact do learn latent historical information when being trained on cognate prediction in a multilingual setup. To support this work, I also created a lexicon generator to create artificial data following realistic language change rules, as well as updated an etymological database to generate cognate datasets to use.

In part I, we travel through the basic concepts needed to understand this thesis. Chapter 1 is about the word from a linguistic standpoint; we therefore first explore what a word is or could be, what linguistic properties it carries, as well as its function as a unit. We then look at how words change, and how these changes are modelled, focusing on historical word prediction, our task of interest. Chapter 2, on the other hand, is about neural networks and their mathematical definitions. We first introduce the basic behavior of a neural network, then follow a panorama of useful neural components for text, from units to full architectures. These two chapters have different audiences: chapter 1 is destined for engineers or natural language processing people wanting to understand more about the linguistic aspects of this task, whereas chapter 2 is destined for linguists wishing to understand neural networks 'from scratch': none of these chapters assume prior knowledge in either linguistics or NLP. Chapter 3 goes through the machine learning pipeline, to provide an overview of the experimental steps, from gathering data and designing experiments to training, validating, and interpreting models. Chapter 4 focuses on the task from which we gather inspiration in this document: low-resource machine translation: it first introduces machine translation, then several tools and techniques specific to situations with scarce data, which can otherwise undergo a drop in performance.

Part II contain the core of the thesis. It first introduces the detail of our research hypothesis and questions, then chapter 5 presents the different neural architectures and datasets we will use to answer said questions. Chapter 6 details the experiments we designed, using artificial datasets, to determine whether and which neural networks can be used for historical word prediction, and under which conditions. Chapter 7 extends these experiments to real world datasets, constituted for the occasion; it also contains a panel of tools and techniques tried to improve model performance. Chapter 8, using previously studied best performing models, focuses on understanding what they learn when training on our task, using both external and internal probes.

Part III is an opening section, where we try to apply our methods to other units in Chapter 9, to study lexicon induction for dialectal variation.

Lastly, we summarize the work done and open new perspectives in part 9.6.

Note Other things were also done during this thesis, and will not be mentioned here in detail:

- the full implementation of a composite multilingual neural architecture for cognate prediction called MEDeA (Multiway Encoder Decoder Architecture), combining multilingual attentional recurrent, convolutional, Transformer-based and siamese encoder-decoders through a fixed size hidden representation. It became the basis of CopperMT (Fourrier et al. 2021), which uses its reimplementation of recurrent encoder-decoders with attention;
- experiments on managing phonetical embeddings, which found that initialising phonetic embeddings with pre-defined features extracted from the litterature actually proved less effective than learning embeddings along the model from a random initialisation;
- published work on the use of modern language models to do named entity recognition on historical text for the BigScience working group (De Toni et al. 2022);
- the complete update of an etymological database, as well as the publication of a detailed methodology to help similar work (Fourrier and Sagot 2020b);
- published work on the use of cognates and borrowings to study semantic change (Fourrier and Montariol 2022);
- an internship at HuggingFace to work on using graph data with transformers;
- outreach activities for high-school students: I gave three conferences on research and NLP, organised two 2-days outreach events (2019, 2020), took part in nine speed-meetings with girls to present computer science jobs — I was also invited to talk about the impact of outreach on gender stereotypes at the French *Assemblée Nationale*;
- teaching Python and programming at the *École Nationale des Ponts et Chaussées*.

Part I

THEORETICAL BACKGROUND

1 Historical words

I won't use words again
They don't mean what I meant
They don't say what I said
They're just the crust of the meaning
With realms underneath.

Vega (1987)

In this thesis, we study the change of words through space and time, focusing on the tasks of historical word reconstruction and dialectal variation study. Assuming that the concept of 'word' exists and is already defined,[1] we will first look at some of its properties (with a focus on the one most interesting for our task: phonology). As we are studying word change, we will also introduce concepts specific to word change study, such as sound change regularity, which constitute the underlying theoretical aspect of our task.

1.1 Synchronic properties of words

1.1.1 Phonology

Phonotactics studies phonological rules, and defines the allowed arrangement of sounds and sound patterns in a language. For example, in the margin notes poem, almost none of the words exist but they all seem intuitively real because they fit English phonotactics. Phonotactics, in a sense, is a 'sound syntax'. **Phonology**, complementarily, studies sound organisation in languages. Distinct language sounds (sometimes gestures) are called **phones**, and they constitute the 'absolute units' of phonetics. These phones are often graphically represented using the International Phonetic Alphabet (**IPA**), which attempts to represent all the distinctive sounds of the world's language.

'Twas brillig, and
the slithy toves
Did gyre and gimble
in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.'
Caroll (1865)

Note: The IPA cannot 'symbolize small, language specific, differences in parameters, [...] does not have distinct symbols for some really distinctive sounds' (Ladefoged 1996), and it relies on theoretical conceptualizations 'sometimes at odds with the physical speech event' (International Phonetic Association 1999). This led to the introduction of many systems of phonetic notation by linguists, which Anderson et al. (2018) have attempted to homogenise to a broader version of the IPA in their CLTS database.

[1] This is a strong assumption but discussing the existence of the word as a unit is outside of the scope of this manuscript.

Two phones which can be substituted by one another without changing word meaning are called **allophones**, such as the sounds corresponding to the letter r in French, pronounced [ʁ] in general, and sometimes [r] in the south of France, which does not change the meaning of words. On the contrary, phones which constitute a minimal pair changing the meaning of words when switched are called **phonemes**: for example, in French, [v] and [b] are phonemes (differentiating between *vin* [vɛ̃] 'wine', and *bain* [bɛ̃] 'bath'), but [ð] (first sound in the English 'this') is not, as it is not used in any French word.

Phonetically, phones are organised in two main groups, depending on the air flow during pronunciation: if the flow is unconstricted, the phone is a **vowel**, else a **consonant**. Consonnants can be pulmonic (air stream from the lungs), glottalic (air stream from the glottis), lingual (air stream from the mouth), or percussive (no air stream). Phonology adds the extra constraint that vowels must form the peak of a syllable; this creates a third category of sounds considered as consonants in their positionning (beginning/end of syllables) but as vowels in their pronunciation, such as the sound [w] in English 'well' or French 'oui': they are called **semivowels** or **approximants**.[2]

Vowels are defined using features, among which the following.

- **Height** is the up/down position of the tip of the tongue in the mouth during pronunciation (e.g. the vowel in English *cat* [kat] is lower than the vowel in English *kit* [kɪt]).
- **Backness** is the front/back position of the upper part of the tongue in the mouth during pronunciation (e.g. the vowel in English *pot* [pɔt] is more back than the vowel in English *pit* [pɪt]).
- **Length** is how long the vowels are (e.g. the vowel in French *mettre* [mɛtʁ] 'to put' is shorter than the one in Frenh *maître* [mɛːtʁ] 'master', though the difference is slowly disappearing in metropolitan French).
- **Rounding** is how rounded the lips are during pronunciation.
- **Nasalization** is linked to the path or the air flow during pronunciation, and whether it goes through the nose or not (e.g. French contains 20% of nasal vowels in its phonetic inventory - and more in some regional variations: [ɛ̃] in *un* 'one', [ɔ̃] in *onze* 'eleven', [ã] in *manger* 'eat').

The two main features of phonetic vowels, height and backness, define a vowel diagram (Figure 1.1), sometimes called the **vocalic triangle**, grouping sounds in three poles: open vowels (a), close-front vowels (i/e), close-back vowels (u/o).

Pulmonic consonants are defined using features, among which the following.

- **Manner** is related to the disruption of the air flow during consonant pronunciation: plosive (e.g [p]) completely interrupt the air flow during pronunciation, fricatives (e.g. [f] or [z]) partially restrain the air flow, and nasals (e.g. [n] or [m]) redirect the air flow to the nose.
- **Place** is linked to the part of the mouth used to pronouce the consonant: labial (e.g. [b]) use the lips, dorsal (e.g. [k]) use the back of the mouth using the palate, and coronal (e.g. [s]) use everything in between.
- **Voicing** or **phonation** is whether the consonant is pronounced with the use of vocal cords or not (e.g. constrast voiced consonant [v] with its unvoiced counterpart [f]).

[2] See Appendix A.1.2 for a list of vowels of interest, and Appendix A.1.1 for a list of pulmonic consonants of interest.

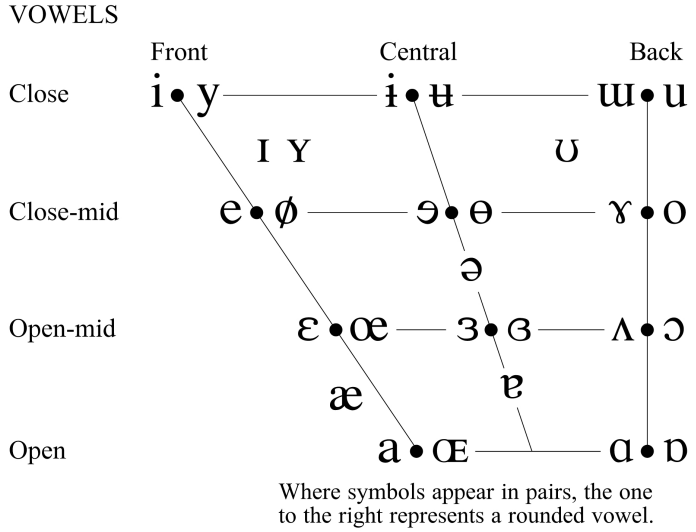


Figure 1.1: Official IPA vowel diagram (CC BY-SA 2020 IPA)

- **Length** or **gemination** is how long the consonant is held (e.g. contrast long 't' in Italian *gatto* [gatːo] 'cat' and short 't' in Spanish *gato* [gato] 'id.').

The two main features of phonetic pulmonic consonants, manner and place, define a consonant chart, see Figure A.1 in Appendix.

Note: When going from oral form to written form, phonemes can be represented through graphic symbols roughly approximating them, letters. This is far from being an universal correspondence, as some languages are spoken only, and others use logograms (Chinese) or syllabic characters (Japanese) to represent the words of their languages - however, for the Romance languages we study, this phone-to-letter mapping is quite regular.

1.1.2 Morphosyntax and semantics

Semantics is the study of word meaning and word relations with respect to meaning, and **morphosyntax** of words and their relations 'in terms of [their] grammatical properties' (Matthews 2007). These fields study **lexemes**, abstract units of meaning, which are associated with an ensemble of **forms**, possible shapes they can take: for example, 'runs', 'running' and 'ran' are forms of the same lexeme, the same concept. Lexemes are also matched with a part-of-speech (**POS**), their grammatical category: the example lexeme is grammatically a verb. Refering to a lexeme is done using a specific form, called the **lemma** (or citation form - literally the associated key in the dictionary), which is for our example 'TO RUN'.

Word forms are made of one or several **morphemes**, minimal units of meaning. The 'root' morpheme which carries the lexeme's meaning is called **stem**: in 'running', the stem is 'run'.[3] Then, from this stem, new words or forms are created by adding **affixes** (attached morphemes) through two mechanisms. **Inflection** creates new word forms for a given lexeme, to reflect its grammatical behavior in context (such as during conjugation to fit the tense, voicing, gender,

[3] In coumpound words, stems can be more complex but this is out of scope.

number): 'running' is an inflected form of 'TO RUN'. **Derivation** actually creates a new lexeme in another grammatical category: 'runner' is a derived form which corresponds to a new concept, a new lexeme.

1.2 Some aspects of words in diachrony

Now that we have browsed some important properties of the word at a given point in time, we will look both at the tools and theory studying word change.

1.2.1 Theory

Etymology is the field which studies the evolution of words, semantically and phonetically, through time and space. This aspect of linguistic is central to the first task of interest of this document, cognate prediction, and it relies on a specific aspect of word evolution: sound change.

1.2.1.1 Phonological change

Phonetic evolution of words is determined using the comparative method, which relies on the Neogrammarians hypothesis of **the regularity of sound changes** (Osthoff and Brugmann 1878): if a phone in a word, at a given moment in the history of a given language, evolves into another phone, then all occurrences of the same phone in the same phonetic context in the same language evolve in the same way. For example, initial [ka] in Latin regularly evolved into [ʃa] in French, which is a pattern appearing in *castellum* [kastelːum] 'castle' evolving into *château* [ʃɑto] 'id.', *cattus* [**k**atus] 'cat' into *chat* [ʃa], and *calor* [kalor] 'warmth' into *chaleur* [ʃalor] 'id.'. Sound changes have been identified, as in the above example, by looking for regular patterns in the attested (or hypothesised)[4] phonetic form of related words.

Some of these related words are called **cognates**:[5] given languages with a common direct ancestor, words are said to be cognates if they are an evolution of the same word, called their **proto-form**. For example, Galician, Portuguese and Spanish *gato*, Catalan and Occitan *gat*, Italian *gatto*, French *chat* and Aromanian *cătushi*, all meaning 'cat', as well as Romanian *cătușă* meaning 'manacle',[6] all descend from the same word *cattus* 'cat' in their mutual parent language, Latin. These cognates underwent the characteristic sound changes of their respective languages' evolution from the parent form to the current state of the language, and have become characteristic of these patterns (Hauer and Kondrak 2011; List et al. 2017).

Comparing the phonetic form of sets of cognates allows to identify some of these patterns:

[4] For dead languages, pronunciation is obviously not obtainable from native speakers, and is therefore reconstructed. Pronunciation guides, poetry rules, or proper name pronuciation 'translation' in other languages are some of the existing third-party sources which can help this reconstruction (Fortson IV 2011).

[5] Outside of historical linguistics, cognates have been defined more broadly as words sharing spelling and meaning, regardless of their etymology (Frunza and Inkpen 2006; Frunza and Inkpen 2009), or words etymologically related no matter the relation (Hämäläinen and Rueter 2019).

[6] In Aromanian and Romanian, the words also underwent diminutive suffixes (*-ushi* and *-ușă*) additions to the now lost cognate root.

in our example, initial [g] in Galician to Italian corresponds to [ʃ] in French and [k] in Romanian and Aromanian. A good explanation of the comparative method and its applications can be found in (Fortson IV 2011).

When a word is an evolution of a form from a language which does not belong to its direct ancestors, it is called a **borrowing** or **loan-word**. For example, English *bonus* is a borrowing from Latin *bonus*, as the Latin word is its origin, though English does not descend from Latin. Borrowings mostly occur to designate 'realities that were unknown before the adopting speech community got in contact with the "giving" culture and its language' or to replace already existing meanings by the word of the related dominant culture (Krefeld 2013)

1.2.1.2 Semantic change

Semantic change does not have the regularity of sound change, as the former rarely affects groups of words together; however, general tendencies can be identified in languages, such as narrowing or broadening of meaning, metaphor, change of positive/negative associations, and so forth (Durkin 2015).[7] Studying semantic change is therefore based on the comparative method and the regularity of sound changes to identify the different forms that words have taken in their history, combined with corpus analysis to understand the evolution of their meanings.

1.2.2 Modeling change through time

Cognate prediction tries to product likely cognates in related languages, and attempts to fill cognate sets by generating plausible missing forms; this lexical task models the regular, word-internal sound changes that transform words over time. It is linked to **cognate identification**, which attempts to find and cluster cognates in a multilingual word sets, and to **protoform reconstruction**, which tries to determine the likely ancestor of cognates, based on the sound changes applied to their respective languages and the phonology and phonotactics of the parent language.

Note: Works on cognates tend to use one of the following definitions, depending on the tasks and units, where cognates can be:

- 'true friends' (words similar in shape and usually meaning, such as English *angel* and French *ange* 'id.') with no regard for their etymology (in the language learning field and sometimes translation), for example, in cognate identification (Frunza and Inkpen 2006; Frunza and Inkpen 2009; Frunza 2006; Inkpen et al. 2005; Kanojia et al. 2020; Kanojia et al. 2021a; Kanojia et al. 2021b; Kondrak 2001; Kondrak et al. 2003; Lefever et al. 2020; Mann and Yarowsky 2001; Markov et al. 2019; Mitkov et al. 2007; Navlea and Todiraşcu 2011; Saadane et al. 2012; Sepúlveda Torres and Aluísio 2011; Sitbon et al. 2015), cognate prediction (Babych 2016; Beinborn et al. 2013; Čulo and Nitzke 2016; List et al. 2022; McCoy and Frank 2018; Mulloni 2007), phonetisation (Nguyen et al. 2018) or lexicon induction (Hauer et al. 2017; Scherrer 2008);

[7] We do not mention lexicogenesis here, but new words created from existing words (by appearing in another language through borrowing mechanisms or evolving in the same language through morphological processes, such as derivation or composition) can also change in meaning when compared to their source.

- words sharing a common ancestor, actually grouping both our previous definitions of cognates and borrowings, for cognate identification (Bloodgood and Strauss 2017; Dinu and Ciobanu 2014; Hauer and Kondrak 2011; Kondrak 2002; Kumar et al. 2017; Rabinovich et al. 2018; Rama 2015; Soisalon-Soininen and Granroth-Wilding 2019; Wu and Yarowsky 2018), cognate prediction (Ciobanu et al. 2020; Hämäläinen and Rueter 2019; Wu and Yarowsky 2018), script decyphering (Luo et al. 2021) or semantic analysis (Frossard et al. 2020; Uban et al. 2021; Uban et al. 2020);
- words sharing a common ancestor which belongs to their direct ancestry, which excludes borrowings,[8] and is the previous definition we introduced, for example in cognate identification (Ciobanu and Dinu 2014; Ciobanu and Dinu 2019; Hewson 1973; Jäger et al. 2017; List 2012; List and Forkel 2022; List et al. 2017; List et al. 2018; Rama and List 2019; Rama et al. 2018; Sims-Williams 2018; St Arnaud et al. 2017), cognate prediction (Bodt et al. 2018; Bodt and List 2019; Ciobanu and Dinu 2019; Dekker 2018; Hall and Klein 2010; Hall and Klein 2011; Nitschke 2021), protoform reconstruction (Bouchard et al. 2007; Bouchard-Côté et al. 2009; Ciobanu and Dinu 2019; Hall and Klein 2011; Hartmann 2019; Hewson 1973; List et al. 2022; Meloni et al. 2021; Sims-Williams 2018), phylogeny (Dekker 2018; Greenhill 2011; Jäger and List 2016; Rama et al. 2018), or borrowing detection (Ciobanu and Dinu 2015; Ciobanu and Dinu 2019; List and Forkel 2022).

The methods introduced here therefore concern either task, but the thesis itself looks at cognates as etymological items, following the last definition.

1.2.2.1 Automatic cognate tasks

Over the last three decades, **automatic cognate identification** has benefited from advances in computational techniques, first using rule-based algorithms (Hewson 1973), transducers (Mann and Yarowsky 2001) or dictionary-based methods (Ciobanu and Dinu 2020; Dinu and Ciobanu 2014; Rabinovich et al. 2018) and similarity metrics (Babych 2016; Frunza and Inkpen 2009; Inkpen et al. 2005; Kondrak 2001; Kondrak et al. 2003; Mann and Yarowsky 2001; Mitkov et al. 2007; Navlea and Todiraşcu 2011; Saadane et al. 2012), then similarity metrics combined with graph or classifiers, followed by clustering algorithms (Bloodgood and Strauss 2017; Ciobanu and Dinu 2014; Hauer and Kondrak 2011; Jäger et al. 2017; List 2012; List et al. 2017; List et al. 2018; Rama 2015; Rama and List 2019; Soisalon-Soininen and Granroth-Wilding 2019; St Arnaud et al. 2017), or similarity metrics directly provided to neural classifiers (Frunza and Inkpen 2006; Frunza 2006; Kanojia et al. 2021b; Lefever et al. 2020; Markov et al. 2019; Sepúlveda Torres and Aluísio 2011) and neural classifiers based on siamese-like architectures (Kanojia et al. 2020; Kumar et al. 2017).

Proto-form reconstruction synonyms: Ancestor reconstruction, ancestor induction, proto-language reconstruction, phonological reconstruction, etymon reconstriction, backward reconstruction.
Cognate prediction synonyms: Cognate chain completion, sister lexicon induction, cognate induction, retroflex generation, reflex retrodiction, forward reconstruction, inference of sound correspondence patterns.

Automatic cognate prediction has been less studied, despite its interesting applications, such as predicting plausible new cognates to help field linguists (Bodt et al. 2018; Bodt and List 2019) and inducing translation lexicons (Mann and Yarowsky 2001), and is often jointly considered, with **automatic proto-form reconstruction**, as an **historical word prediction** task (an expression coined by Dekker (2018)). They have been approached with rule-based systems (Heeringa and Joseph 2007; Marr and Mortensen 2020; Nasution et al. 2016; Smith 1969), phylogenetic trees combined with stochastic sound change models (Bouchard et al. 2007; Bouchard-

[8] French *ange* and English *angel* are not cognates according to this last definition, as though both descend from Latin, Latin is not a direct ancestor of English, and *angel* is therefore a borrowing.

Côté et al. 2009; Bouchard-Côté et al. 2013; Hall and Klein 2010; Hall and Klein 2011), purely statistical methods (Bodt et al. 2018), similarity and clustering approaches (Bodt et al. 2018; Bodt and List 2019; Ciobanu and Dinu 2015; Greenhill 2011; Hauer et al. 2017; List et al. 2022; McCoy and Frank 2018), neural networks (Luo et al. 2021; Mulloni 2007; Nguyen et al. 2018; Wu and Yarowsky 2018), language models (Hauer et al. 2019) and character-level machine translation techniques (Beinborn et al. 2013; Dekker 2018; Hämäläinen and Rueter 2019; Meloni et al. 2021; Nitschke 2021), because of their similarity to a translation task (modelling sequence-to-sequence cross-lingual correspondences between words).[9]

An exhaustive survey of historical word prediction methods specifically has been published by Dekker and Zuidema (2021), and a broader one on historical phonology related tasks by Sims-Williams (2018).

1.2.2.2 Databases

To work on historical word reconstruction, we need cognate sets; historically, the first ones come from the Swadesh (1955) lists, manually annotated correspondences between related languages (containing between 50 and 200 word pairs for the available languages). Yet, manual cognate sets tend to be costly, and are not necessarily digitized. The following etymological databases, automatically extracted from online sources, combine large scope, usable data format, and generally reliable sources. Interestingly, a majority of these large scale electronic etymological databases have been generated from a version of the Wiktionary, an online collaborative dictionary, which contains large scale structured information, most of the time sourced from already existing and published etymological works. These databases mostly restructure existing information from wordnets or the Wiktionary, notably to extract 'longer range' word relations.[10]

EtymWordNet (Melo 2014) is an etymological database extracted from the 2013 version of the Wiktionary. It makes a difference between cognacy (in its strictest sense) and generic "etymological origin" relations, but goes no further. It also does not systematically differentiate between glosses.[11] It contains 473,433 general etymological relations and 538,588 cognacy relations. **CogNet** (Batsuren et al. 2019; Batsuren et al. 2021) is an automatically extracted cognate database based on wordnets. It uses a loose definition of cognacy, and therefore is actually a database containing both cognates and loanwords. It has the lowest granularity, but the most lexemes, with 3 million "cognate pairs" across 338 languages. **EtymDB 1.0** (Sagot 2017) is an etymological database automatically extracted from the Wiktionary. It makes a difference between inheritance, borrowing, and cognacy, and contains 1 million distinct lexemes linked by half a million distinct relations. However, it has a few shortcomings, most notably in its management of duplicates.[12] **CoBL** (Anderson et al. 2020), the Cognacy in Basic Lexicon database is a descendant and upgrade of Michael Dunn's Indo-European Lexical Cognacy Database, and though it only concerns itself with cognacy, its sources have been handpicked, but it is still not public yet.

[9] Our own work (Fourrier et al. 2021; Fourrier and Sagot 2022) follows the character-level MT approach, as we will present in this thesis.

[10] For example, if A is an child of B, and B a child of C, therefore C is an ancestor of A, but the relation from C to A might not be explicit in the source, and is made explicit in the extracted database.

[11] Here, the gloss of a word refers to its meaning expressed as its English translation.

[12] These are shortcomings we adressed in our upgrade of the database, see Fourrier and Sagot (2020b).

It is also possible to do a manual extraction of the Wiktionary directly using an existing script, as provided in **EtymDB 1.0**, (Wu and Yarowsky 2020) and (Pantaleo et al. 2017). **EtyTree** (Pantaleo et al. 2017) also contains a tool to visualise the Wiktionary as a graphical etymological dictionary.

1.3 Conclusion

As we saw in this section, it is very hard to define what a word is, though specific properties, such as phonetic ones, can be studied. However, as we study words not in isolation, but through time and space, we introduced the theoretical tools necessary to undertake these tasks. Of these, historical word prediction, or studying 'word' change through time, is the first axis of interest of this thesis. As seen above, many automatic methods can be applied to this task, but we are specifically interested in neural networks; this had barely been done before the beginning of this thesis (Beinborn et al. 2013; Dekker 2018), with at the time extremely mitigated results. We aimed to challenge this, and exhaustively study the applicability and usefulness of neural networks for this task. But first, before going into the details of our experiments, we must first understand how neural networks work.

2 The mathematics of neural networks for text

> I may remark that the curious transformations many formulae can undergo [...] is I think one of the chief difficulties in the early part of mathematical studies. I am often reminded of certain sprites and fairies one reads of, who are at one's elbows in one shape now, and the next minute in a form most dissimilar.
>
> Lovelace (1840)

Natural language processing is at the intersection of linguistics, computer science, and mathematics. To analyse and process natural language data, the field uses, among other techniques, machine learning tools called artificial neural networks (frequently abridged to simply 'neural networks'). In this chapter, we explore their general behavior and life cycle, then focus on some neural networks of interest, frequently applied to written text.

The mathematical notations chosen in this chapter have been homogenised across papers for consistency and easier algorithm comparison.

2.1 How do neural networks work?

2.1.1 First implementation of an artificial neural network

The first implementation of an artificial neural network is the perceptron (Rosenblatt 1958),[1] and this work, though no longer state-of-the-art, can help understand the priors behind the design of such structures. The author first presents two theories of how the brain possibly works (based on the biological knowledge of the time) regarding how information is sensed, stored, then affects behaviour through memory. He subsequently proceeds to choose one as his starting point, the connectionist approach, which supposes that information is stored in the brain as connections and pathways, instead of stored in fixed locations. Rosenblatt expresses this as postulates about the plausible behaviour of biological neural networks, which then translate to programmatic priors for artificial neural networks:

1. A biological neural network is random at the birth of its owner.
   → An artificial neural network is initialised randomly.

[1] Theoretical foundations had been laid before, for example in Turing (1948).

2. The brain is plastic: connections change through time, and connection formation is impacted by positive or negative reinforcement.
   → The artificial network's connections change through time, depending on (mathematically) positive or negative reinforcement.
3. Similar stimuli generate similar electric paths through similar biological neurons.
   → Similar inputs generate similar connections through similar artificial neurons.

This leads him to define the perceptron as a series of connections between:

- 'sensory units': they activate depending on the stimuli they receive, mimicking for example visual or touch sensory neurons,
- 'association units': they activate based on the weighted sum of the input they receive from the sensory units, mimicking relay neurons,
- 'response cells': they receive information from the association units, and can send backward feedback, mimicking the response given by motor neurons.

In modern versions of the perceptron, these units are simply called **neurons**, grouped in successive **layers**: the perceptron is made of three layers, the input/hidden/output layers, which respectively contain the sensory/association/response neurons. The connection weights and neuron inner values (biases) are initialised randomly at first (following postulate 1).[2]

To teach the model to learn to store and predict information, it must be trained: when presented with a stimuli (input), the perceptron outputs an answer (through the association and response cells), and is then updated using positive or negative reinforcement (from the response cells), to encourage or inhibit the connections based on the adequacy between actual received stimuli and ideal expected response.[3] This training is done for a specific task, classification in the paper: the model is trained to associate an input stimuli with a given class (for example an image with a label). When it has learned, the model must be frozen to evaluate its performance: this is called testing. Rosenblatt, in his paper, introduces two evaluations: recall is computed on the same data (to see how much information has been retained),[4] and generalisation on data never before seen by the model (to see how well the model can generalize what it has learnt on new examples).

These training experiments are repeated while varying model parameters[5] (threshold for activation, number of connections...) to find the best ones for the classifying problem at hand.

2.1.2 Life of a supervised neural network

Supervised neural networks tend to follow the same life-cycle: conception, training, evaluation, parameter choice, with some other steps, such as validation or fine-tuning (see below).

[2] Artificial neural networks were historically *inspired* by biological neural networks at first, but they are not expected to model them, nor to provide insights on how the brain works.

[3] The update by positive or negative reinforcement is a simple version of 'backpropagation,' see Section 3.

[4] This can seem extremely counter intuitive with respect to our modern vision of machine learning, where training data is *never* used for testing.

[5] These are usually called hyper-parameters in machine learning lingo.

Conception Creating a neural network means defining its topology/architecture (neurons organisation and their connections), initialisation (initial value of all weights and biases), and the learning rules it follows (how error is computed, how the positive and negative reinforcement is provided...). The topology is usually chosen to obey priors (biologically-inspired priors in the case of the perceptron) or solve the problems introduced by a specific task (such as learning ordered sequences, which we will see in Section 2.2) for which data is available or has been created.

Training The neural network is then trained: from a provided input, it predicts an output, compared to the reference (also called gold, or target) using a function called the **loss function** (or error/cost/objective function). The goal is to minimise the error, the difference between model prediction and actual target. This is done using **optimisers**: they change slightly the inner values of the model (weights, biases) to make them tend towards what the optimal values would have been to get a minimal error for the current input/output association. This process is called **back-propagation**, and how much the inner values change depends on the **learning rate** (the size of the change step). This training loop is then repeated for all data examples available; an **epoch** has happened when the model has seen all the data available one time. Data can be provided to the model all at once (full **batch**), by grouping examples into **mini-batches**, or one example at a time (mini-batches of size one). Training is usually stopped after multiple epochs, when the loss function has converged (the error has stopped diminishing).[6]

The litterature is not unanimous on its use of the terms **batch** and **mini-batch**. Many papers actually use 'batch' to mean 'a subgroup of the total examples provided at once,' what we call a mini-batch; mini-batch is usually used as 'the subgroup of items before a back-propagation step'.

Hyper-parameters tuning The training can also be replicated several times using different combinations of the training parameters values (number of layers, size of layers, input representation size, learning rate...) to choose the best combination for the current model and data.

Testing At the end of the model training, its inner values are frozen, and new data is provided (**a testing set**), for which it has to predict associated targets from the inputs. The number of correctly predicted targets gives the accuracy of the model, and the overall process is called testing. It indicates how well the model generalises to new, unseen data.

Validation To prevent our neural network from learning its data 'by heart,' which is called **over-fitting**, it can be regularly (e.g. at the end of each epoch) tested on data which is not in the training (nor usually testing)[7] sets (development or **validation data**): this is called validation. Usually, at the start, the validation accuracy increases jointly with the training accuracy. However, after a number of epochs, when the validation accuracy diminishes but not the training accuracy, it indicates that the model starts to over-fit its training data, and loses its ability to generalise. It can therefore be a good moment to stop training.

Fine-tuning A trained model can be then specialised on a new task, by allowing its inner weights and biases to be updated once more, but using a new loss function according to a new

[6] It can also happen that the loss function does not converge, in which case other stategies must be tried.

[7] Since we use the validation set to choose a 'best' model, we do not want validation data to intersect with testing data, as testing must highlight generalization to new data.

training objective and a new task, with new data, and sometimes even new model layers to train along the new loss. This operation can be seen either as specializing an existing model on a new task (**training, then fine-tuning**), or choosing a good initialisation for a model before training it on a given task (**pretraining, then training**).

2.1.3 Feed Forward Neural Network

Now that we saw what the general lifecycle of a neural network looks like, we focus on how it translates to mathematical equations, looking at conception and training, and using as example a more complex and general form of the perceptron: the Feed Forward Neural Network (FFNN).[8]

2.1.3.1 Conception

Simple perceptrons can only be used as linear classifiers (they can only map inputs to binary categories). However, increasing the number of hidden layers, which creates multi-layered perceptrons (also called fully connected FFNN), has been shown to transform them into universal function approximators (Hornik 1991; Hornik et al. 1989). Universal function approximators can theoretically learn any pattern between input and output in a data set, for example for classification (mapping input values to a finite number of classes, such as mapping words to their part of speech) or for regression (mapping input values to a continuous output space, therefore a possibly infinite number of values, such as mapping a day of the year to its temperature).

Our general FFNN can be expressed as the equations of its three layer-types neurons.[9]

- For the input layer, $L^{(0)}$, the output of the i-th neuron is:

$$net_i^{(0)} = S\left(\sum_{j=1}^{\#X} W_{ij}^{(0)} x_j\right) \tag{2.1}$$

  All the different inputs x_j (from the total input list X) are connected to the neuron i through weights $W_{ij}^{(0)}$. The output of neuron i is the weighted sum of all the received inputs, passed through a threshold function (often a sigmoid S, which fits values between 0 and 1), which might be changed for different layers. The threshold function is also called an **activation function**, and it is inspired by the activation potential of a biological neuron: if the electrical potential reaching a neuron passes a threshold, the neuron activates.
- For the l-th of z hidden layer, $L^{(l)}$, the output of a neuron i is:

$$net_i^{(l)} = S\left(\sum_{j=1}^{\#L^{(l-1)}} W_{ij}^{(l)} net_j^{(l-1)}\right) \tag{2.2}$$

  and

[8] To simplify the equations of this subsection, we ignore the biases.

[9] A multilayered perceptron is a specific implementation of the fully connected feed forward neural network, with at least one hidden layer, and none of its weights and biases maintained null. For computation simplicity, however, feed forward neural networks are rarely entirely connected.

- For the output layer, $L^{(z+1)}$, the output of a neuron i is:

$$\hat{y}_i = S\left(\sum_{j=1}^{\#L^{(z)}} W_{ij}^{(z+1)} net_j^{(z)}\right) \tag{2.3}$$

as hidden and output layers behave like the input layer, using the output of their previous layer as input.

2.1.3.2 Training

Once a model has been defined, it can then be trained to learn patterns in our data. Training, on a given sample X, happens in three steps.

1. **Forward propagation**
   The given input goes through the full network function, which outputs a prediction:[10] its candidate target for the input, $\hat{Y} = network(X)$.
2. **Loss computation**
   The network computes the difference between its prediction $\hat{Y}$ and the original target Y. The loss function depends on the chosen task and model: for a regression task, the mean squared error (MSE) is often chosen, whereas a classification task would typically use a cross entropy loss (CEL): mean squared error computes the difference between predicted and target *values*, where cross entropy loss computes the difference between predicted and target *distributions*, i.e:

$$\mathrm{E} = \sum_{i=1}^{\#Y} \frac{1}{2}(\hat{Y}_i - Y_i)^2 \tag{2.4}$$

   .
3. **Backpropagation**
   The goal of backpropagation (term coined by Rumelhart et al. (1986)) is to recursively update the model weights to minimize the global error (or loss), using an optimizer, gradient descent in the original paper.[11]

   Optimizers therefore travel along the studied function by following the downward direction of their slopes, given by their derivative functions. To find the error minimum, we need its derivative. However, this error depends on the weights: we compute the partial derivative of the error with respect to each weight: these partial derivatives represent how the the global error is affected by each weight, or the 'feedback' of each weight to the global error. We then change each weight by an amount proportional to these derivatives. As each layer depends on the previous one, the computations must be done hidden layer per hidden layer, starting from the end ones (this is called the **chain rule**).
   In our example, applying the chain rule to the partial derivative of the error with respect

Finding a function minimum[12] means looking at the function shape to find its lowest point. Mathematically speaking, the slope of a function at any point is given by its derivative at said point.[13]

[10] Forward propagation is also called forward pass.

[11] Several other optimizer algorithms have been introduced since, such as Adagrad, Adadelta, Adam, RMSProp, and so forth, each with its own convergence speed and robustness. A good comparison can be found at `http://ruder.io/optimizing-gradient-descent`.

[12] The function minimum found might be local.

[13] This only holds for derivable functions, and this can be part of the challenge when designing neural networks and optimizers: some interesting architecture choices might imply using non-continuously derivable functions, which introduce their own set of problems for backpropagation. However, this is completely out of scope for this document.

to a given weight $W_{ij}^{(z)}$ in the last layer z (for output i and node j) gives:

$$\frac{\partial \mathrm{E}}{\partial W_{ij}^{(z)}} = \frac{\partial \mathrm{E}}{\partial \hat{Y}_i} \frac{\partial \hat{Y}_i}{\partial net_j^{(z)}} \frac{\partial net_j^{(z)}}{\partial W_{ij}^{(z)}} \tag{2.5}$$

In more detail, we observe that the partial derivative of the mean squared error with respect to the weight can be computed, since all expressions included in the partial derivatives below can be derived:

$$\frac{\partial \mathrm{E}}{\partial W_{ij}^{(z)}} = \frac{\partial \left(\sum_{i=1}^{\#Y} \frac{1}{2} \left(\hat{Y}_i - Y_i^2 \right) \right)}{\partial Y_i} \frac{\partial \left(S \left(\sum_{j=1}^{\#L^z} W_{ij}^{(z+1)} net_j^{(z)} \right) \right)}{\partial net_j^{(z)}} \frac{\partial \left(S \left(\sum_{k=1}^{\#L^{z-1}} W_{jk}^{(z)} net_k^{(z-1)} \right) \right)}{\partial W_{ij}^{(z)}} \tag{2.6}$$

The new weight value is then updated by an η factor (the learning rate), following:

$$W_{ij}^{(z)} \Leftarrow W_{ij}^{(z)} - \eta \frac{\partial \mathrm{E}}{\partial W_{ij}^{(z)}} \tag{2.7}$$

The weight update is therefore big when the impact of the weight on the error is big. These equations can then be applied to the next layer, adding another link to the chain rule, and so forth, till we went back to the start of the neural network.

When all weights W have been updated (with their contents changed) to minimize the error, the process is repeated until convergence has been reached (the error has been below a threshold for n epochs) or a certain number of epochs has happened. The model is then frozen.

2.2 Managing Sequences: Recurrent Neural Networks

As we saw earlier, FFNNs can model patterns in the data, where a given input point is associated with a given output class or value. However, this does not work with sequential data, where the input items' order must be retained in some way, since the input layer connects to all input points no matter this order.

Recurrent neural networks (RNNs) are derived from FFNNs, but they have been specifically designed to mitigate the problems of temporality/sequentiality in data, to be able to study sequences while retaining information about their items' order: in other words, they have been designed to have a memory of sorts. Since text is sequential by nature, these networks are interesting for NLP.

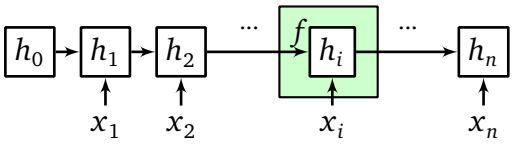


Figure 2.1: Schematic representation of an RNN parsing a sequence.

An RNN models a sequence thanks to an 'internal memory state' mechanism (or 'hidden state' h), initialised at random (h_0). Each item x of the studied sequence is presented successively to the RNN model, which updates its memory state h_i to combine information from the

current item on one hand (x_i), and the previous hidden state on the other hand (h_{i-1}). This allows the model to preserve information 'through time' about items previously seen. This hidden state update (in green on Figure 2.1) can be modelled as:

$$h_i = f(h_{i-1}, x_i) \tag{2.8}$$

However, the interesting feature of an RNN is that, though the internal memory state h_i is updated with each item x_i, the function f used to compute each new internal state (from the previous internal state and current item) is the same for all items of a sequence, and its inner weights and biases are updated at the end of a full sequence, or mini-batch of sequences[14] (using backpropagation): the same model can therefore be used on sequences of different lengths with no adaptation needed, contrary to an FFNN, which has a fixed input size.[15]

Example use cases

1. **Item level labelling:** Some tasks, such as part of speech tagging,[16] need to associate each and every item x_i of a sequence X with a label y_i, while taking into account the context of each item. The model's hidden state h_i at each step can be seen as a representation of the input x_i *in its sequence*, as it carries information from both x_i and all the previously seen elements of said sequence (x_0 to x_{i-1}). We can therefore use these h_is as inputs to train a classifier on the item labelling task of interest.
2. **Sequence level labelling:** Some other tasks, such as sentiment analysis,[17] need to associate a whole sequence X with a single label Y. In this case, we would take the last hidden state of the recurrent model, which carries information from all elements of the sequence, and can therefore be seen as a 'sequence summary,' to train a classifier on the sequence labelling task.

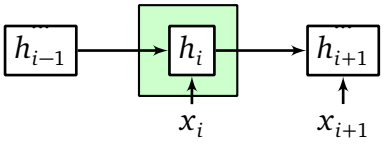
In this section, we will look at different ways to implement recurrent neural networks.

2.2.1 Simple Recurrent Neural Networks

One of the most basic possible RNN instances is the SRNN (Simple Recurrent Neural Network, or Elman network, Elman (1990), Figure 2.2): each hidden state at a given step is a weighted linear combination of itself at the previous time step with the current input, through a non-linear activation function. [18]

This network was shown to be able to store simple sequential information, such as letter order in words, word order in simple sentences (Elman 1990), or even to predict next word POS (Towsey et al. 1998). However, SRNN can sometimes encounter a problem: during backpropagation, the partial derivative of a hidden state i with respect to a previous hidden state k takes the form of a product of $t-k$ matrices, and 'in the same way a product of $t-k$ real numbers

[14] See Sec. 3.1.2 for more detail on batching.
[15] Different update schemes could be considered, such as updating the model after each token of the sequence, but they will not be covered here.
[16] Part of speech tagging links, in its simplest form, each word of a sentence to its grammatical category.
[17] Sentiment analysis classifies the polarity of sentences (positive/negative/neutral) in more or less detail.
[18] Incidentally, it is the default choice when instantiating PyTorch RNN classes.



1. **Hidden state**: $h_i = \tanh(W_h h_{i-1} + V_h x_i + b_h)$

Figure 2.2: Schematic representation of an SRNN layer update step

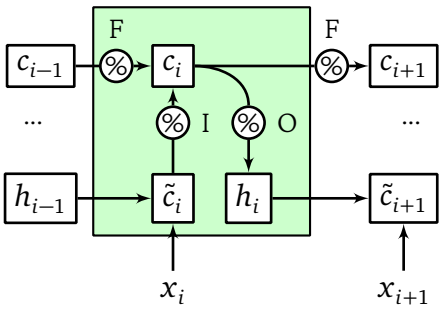
can shrink to zero or explode to infinity, so does this product of matrices' (Pascanu et al. 2013). When the product converges towards zero, it is said that gradients (the partial derivatives) are **vanishing**, and the weight update is null. If it converges towards infinity, gradients are **exploding**, the weight update step is too big and no longer follows the slope. This prevents SRNN from learning long-term information easily, as they can easily stop learning when the gradients are vanishing, or update randomly when the gradients are exploding.

2.2.2 Long Short Term Memory

Hochreiter and Schmidhuber (1997) introduced the Long Short Term Memory unit (LSTM) to prevent vanishing/exploding gradients: the initial design goal behind this new unit was to maintain an constant gradient flow from one time step to the next during back-propagation by altering the mathematical dynamics.

It consists of the following elements:

- a hidden state, reflecting the current state of the cell (as before),
- a cell state (another type of hidden state) to keep more long-term information,
- 'gates,' functions whose role is to determine how much information to keep from their arguments.



1. **Candidate cell state**: $\tilde{c}_i = \tanh(W_c h_{i-1} + V_c x_i + b_c)$
2. **Input gate**: $I = \sigma(W_I h_{i-1} + V_I x_i + b_I)$
3. **Forget gate**: $F = \sigma(W_F h_{i-1} + V_F x_i + b_F)$
4. **Cell state**: $c_i = F \odot c_{i-1} + I \odot \tilde{c}_i$
5. **Output gate**: $O = \sigma(W_O h_{i-1} + V_O x_t + b_O)$
6. **Final hidden state**: $h_i = O \odot c_i$

Figure 2.3: Schematic representation of an LSTM layer update step.
(The input, output and forget gate are all symbolised by '%', as they retain a part of the information they are fed).

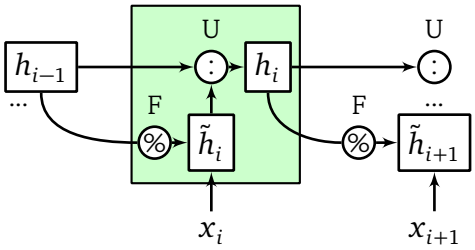
At each step, the current input is combined with the previous hidden state to create a candidate cell state, $\tilde{c}_i$ (1), a new item representing the current and direct past information. The cell state is updated (4) by combining the output of the candidate cell state through an 'input gate' (2) with the output of its previous self through a 'forget gate' (3). The input and forget gates are computed depending on the step's input and hidden state, to learn when to keep input (input

gate) and when to forget previous information (forget gate) depending on said elements: the new cell state is a combination of old (previous cell state through forget gate) and new (candidate cell state through input gate) information selectively retained — this also prevents the cell state from being 'polluted' with irrelevant information, filtered out by the gates. Then, the hidden state is actualised with the cell state (6), going through an 'output gate' (5) which determines, again, the amount of information to keep, depending on the current input and previous hidden state. This succession can be materialised by the logic schematic of Figure 2.3.

The LSTM partially solves the problem of vanishing gradients, while allowing to keep more long term information than a vanilla RNN.

2.2.3 Gated Recurrent Units

In 2014, a simplified version of the LSTM was proposed and called the Gated Recurrent Unit, or GRU (Cho et al. 2014). Its inner components have been reduced to a hidden state to capture information through time, informed by 'gates,' which chose the information to keep.



1. **Forget gate**: $F = \sigma(W_F h_{i-1} + V_F x_i + b_F)$
2. **Candidate hidden state**: $\tilde{h}_i = \tanh(W_H(F \odot h_{i-1}) + V_H x_i + b_H)$
3. **Update gate**: $U = \sigma(W_U h_{i-1} + V_U x_i + b_U)$
4. **Final hidden state**: $h_i = (1 - U) \odot h_{i-1} + U \odot x_i$

Figure 2.4: Schematic representation of a GRU layer update step.
(The forget gate is symbolised by '%', as it retains a part of the information it is fed. The update gate is symbolised by ':', as it keeps a ratio of previous hidden state to newly computed candidate hidden state).

At each step, the candidate hidden state (2) is the combination of the current input with the previous hidden state, which went through a 'forget gate'. This forget gate (1) determines the amount of information to remember from the past. The step hidden state (4) is then updated through a well named 'update gate' (3), which balances the information ratio of 'previous hidden state' to 'newly computed candidate hidden state' to keep. This succession can be materialised by the logic schematic of Figure 2.4.

The GRU contains less internal elements than the LSTM, which makes it lighter to train, while keeping the longer term memory.

2.2.4 Bidirectional recurrent units

All RNNs we saw earlier go from one end of a sequence to the next. However, sometimes, it is interesting to look at a sequence in both directions, and that is when to use a bi-directional RNN (BiRNN). A BiRNN is composed of two RNN units moving respectively forward and backward over the input (Figure 2.5).

The forward RNN (hidden state: $\overrightarrow{h_i}$) reads the input as was described previously, from start to finish. The backward RNN ($\overleftarrow{h_i}$) reads the input from finish to start. The total hidden state (h_i) at a step for a given input (x_i) is the concatenation of the corresponding forward and backward hidden states for said input, one RNN having seen previous words, the other following words. This is interesting when the full context is needed to represent a sequence item.

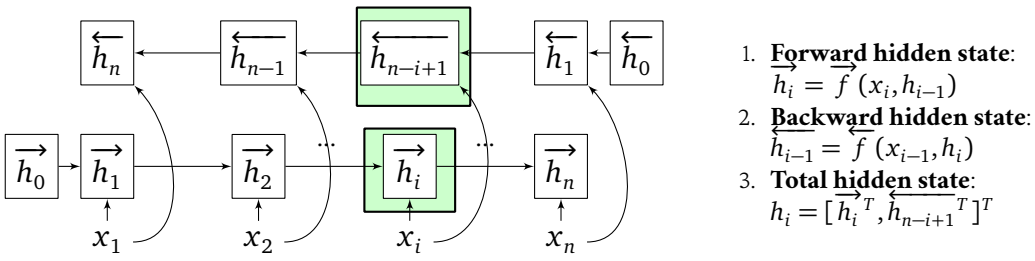


Figure 2.5: Schematic representation of a BiRNN parsing a sequence.

2.3 Many-to-many mapping: Encoder-decoders

2.3.1 Challenge and solution

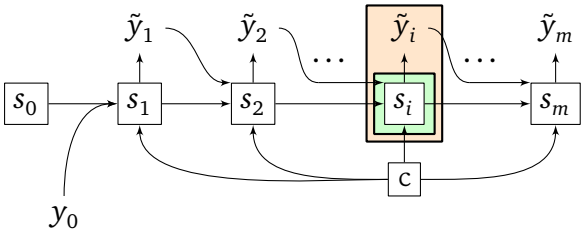
The deep neural networks we saw previously (MLPs, RNNs...) can easily learn a mapping between inputs and outputs in a many-to-one situation (e.g. predicting the label of a whole sentence, as in sentiment analysis) or a one-to-one situation where input and output are aligned (e.g. prediction of labels for every word in a sequence, as in part-of-speech tagging, such as the user cases introduced in Section 2.2). However, they cannot learn, as such, the mapping in a many-to-many situation, where input and output have different sizes and are not necessarily aligned (e.g. translation).

This architecture appears under the names Seq2Seq, sequence to sequence model, or encoder-decoder.

Cho et al. (2014) introduced a method to adress this problem, formalised in the work of Sutskever et al. (2014), who called it 'Sequence to Sequence learning'. A sequence to sequence model could technically be any model which maps a sequence to another sequence, but we study here some of the more well-known approaches to sequence to sequence learning.

The basics of the architecture designed in Cho et al. (2014) and Sutskever et al. (2014) are the following:

- The **input** sequence is mapped, through a neural network called an **encoder**, to a single vector, called the **context** (c, sometimes called the hidden representation).
  In their models, the encoder is an RNN or BiRNN, and the context either the concatenation of all its previous hidden states, or its last hidden state only (since, as we saw earlier, it is a good approximation of the full sequence seen).
- The **context** is then mapped, through another independent neural network called a **decoder**, to a final sequence, the model **output**.
  In their models, the decoder is an RNN, which learns to predicts a sequence of items from two things: the context, and *its own previous predictions*, extracted from its previous hidden states.



1. **Decoder hidden state**:
   $s_i = \text{RNN}([\tilde{y}_{i-1}, c], s_{i-1})$

Figure 2.6: Schematic representation of a decoder step by step.

Figure 2.6 represents the decoding process in more detail. The decoder hidden state is randomly initialised (s_0), and the first sequence item used for prediction (y_0) is a special token, called the beginning-of-sequence token (<BOS>). For its first step, the decoder uses said <BOS> token, its random hidden state, and the encoder context as inputs, which updates its hidden state to h_1. (Where the encoder used the hidden state to represent what it had seen before, the decoder uses the hidden state to represent what it might see next.) Said hidden state represents the probabilities over all possible outputs, and is then converted to the most likely prediction for the current step, $\tilde{y}_1$. This prediction is then provided to the decoder for the next step as if it were the 'actual' next element of the sequence. The decoder also receives the updated hidden state s_1 and the unmoving context c, to update itself. This is repeated till the decoder predicts a special token, called the end-of-sequence token (<EOS>). The apparition of said token stops the decoder, and launches the backpropagation. However, Cho et al. (2014) also showed that the performance of an encoder-decoder decreases considerably for long sentences, as it can be hard for the encoder to compress all needed information in the hidden representation, and then for the decoder to extract relevant information from this representation.

In this section, we study two of the main encoder-decoder architectures used in neural machine translation context.

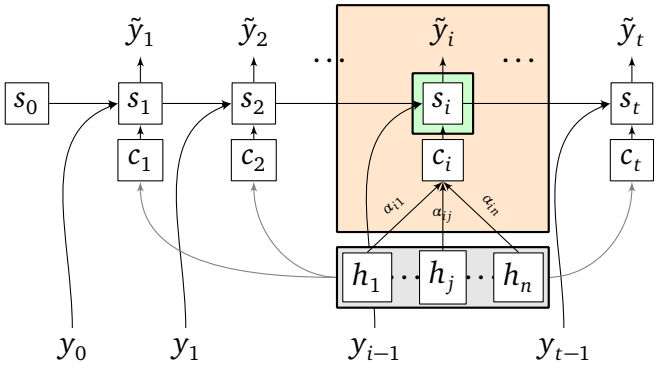
2.3.2 Recurrent Encoder-Decoders

The original architectures of the encoder-decoder (Cho et al. 2014; Sutskever et al. 2014) use, as mentioned before, recurrent units as encoder and decoder.[19]

In parallel, a similar encoder-decoder model was introduced with an additional mechanism: an 'alignement model,' which came to be known later as 'attention' (Bahdanau et al. 2015). The goal of attention was to help the decoder 'focus' on the relevant encoded items for its prediction, by processing the encoder's output to compute a new context at each decoder step.[20]

In more detail (Figure 2.7), at step t, a new context c_t (1) is computed as the weighted sum of the encoder output h, where the weights α_{tj} (2) are the normalized scores of an alignement model (an FFNN) between the previous decoder hidden representation s_{t-1} and encoder output h.[21] This weighted sum can be seen as a probability distribution over the scores of the alignement model. Then, the decoder takes as input its previous hidden state s_{t-1}, the target of the previous step y_{t-1}, and the just computed context c_t, to generate its new hidden state s_t (3), which is

[19] Encoder: reversed LSTM (see Section 2.2.4); Decoder: LSTM.
[20] Encoder: Bi-GRU; Decoder: GRU
[21] The FFNN is trained along the encoder-decoder model.

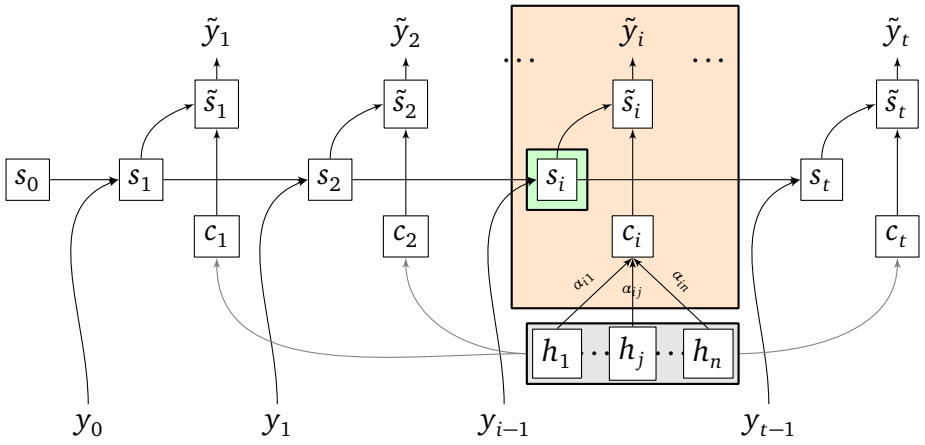


1. **Current context**: $c_t = \sum_{j=1}^{n} \alpha_{tj} h_j$
2. **Context weights**: $\alpha_{tj} = \sigma_j(\text{FFNN}(s_{t-1}, h_j))$
3. **Decoder hidden state**: $s_t = \text{GRU}([y_{t-1}; c_t], s_{t-1})$
4. **Prediction**: $\tilde{y}_t = S(W_s s_t)$

Figure 2.7: Schematic representation of the Bahdanau attention decoding step

then the source of the prediction (4). (We note that here, we do not provide the previous output during training, but the previous actual target: this way, if the output of the previous step was incorrect, the model still has access to the correct previous item for its prediction of the current one. This method is called 'teacher forcing', see Section 2.3.5.)

This attentional mechanism was then simplified (Luong et al. 2015a),[22] as this time, attention is applied to the decoder current hidden state, not to the previous one. The alignement model generates a new 'attentional hidden state,' either globally (looks at the whole context, same as earlier) or locally (focuses on a window of context, which is less expensive than globally), from which to predict outputs. It makes more intuitive sense to predict the alignement of the current item with the context than with the previous item.



1. **Decoder hidden state**: $s_i = \text{GRU}(y_{i-1}, s_{i-1})$
2. **Current global context**: $c_t = \sum_{j=1}^{n} \alpha_{ij} h_j$
3. **Global context weights**: $\alpha_{ij} = \sigma_j(a(s_i, h_j))$
4. **Attentional hidden state**: $\tilde{s}_i = \tanh(W_c[x_i; s_i])$
5. **Prediction**: $\tilde{y}_i = \text{softmax}(W_s \tilde{s}_i)$

Figure 2.8: Schematic representation of the Luong attention decoding step.

In more detail (Figure 2.8), at step t, the RNN updates (1), using its previous hidden step s_{i-1} and target y_{i-1}. A new context is computed, as the weighted sum of the encoder output h components again (2), but this time, the weights α_{ij} (3) are the normalized scores of an alignement model between the *current* decoder hidden representation s_i and encoder output h. Then, this new context c_i is concatenated with the current hidden state s_i and provided to a linear layer, to create an 'attentional' hidden state $\tilde{s}_i$. This attentional hidden state is the source of the predicted output $\tilde{y}_i$ (5).

[22] Encoder: Bi-GRU; Decoder: LSTM

Luong et al. (2015a) also introduced several new ways to compute attention. First, as described earlier in the equations of Figure 2.8, the attention can be computed **globally**, aligning the current hidden state with the whole encoder context at each step, as:

$$\alpha_{ij} = \sigma_j\left(a(s_i, h_j)\right) \tag{2.9}$$

However, global attention, having to attend to all the words, might learn uninteresting patterns, which was at the origin of a more focused **local** attention. In it, the global alignements are weighted with respect to their importance to the currently studied hidden state:

$$\alpha_{ij} = k\sigma_j\left(a(s_i, h_j)\right), \text{with } k = exp\left(-\frac{2(j-p_i)^2}{D^2}\right) \tag{2.10}$$

With j the current position, p_i is the position of the item being aligned with, and D the size of a window around said position. A naive approach can consider that the k must directly depend on the distance to the current item, and use a **monotonic** $p_i = i$, where a more complex approach can try to find the most interesting positions, using a **predictive** $p_i = \#h.\sigma\left(v_p^T \tanh(W_p s_i)\right)$.

Lastly, scores can also be **location-based**, solely dependant on the target hidden state position in the sentence, which does not use an alignement model:

$$\alpha_i = \sigma(W_a s_i) \tag{2.11}$$

Alignement models can also vary, and three have been introduced:

- **dot**: $a(s_i, h_j) = s_i^T h_j$,
- **general**: $a(s_i, h_j) = s_i^T W_a h_j$,
- **concat**: $a(s_i, h_j) = v_a^T \tanh(W_a[s_i, h_j])$.

The alignements are not detailed in the original paper, but the **dot** alignement litteraly computes the relatedness of two vectors, by looking at the product between their respective features. The **general** attention adds a linear layer (W_a), which can learn to weight the most relevant features of the hidden state. Lastly, the **concat** concatenates the hidden state with the encoder representation, and weights the interesting items over both sequences, not necessarily related to one another, then normalizes weights with a tanh layer and reduces dimensions through a vector multiplication.

Why use attention? Attention relieves the complexity of the hidden representation, by allowing relevant contextual information to be spread out: at each step, the model can focus on the context information most relevant to the current target item (via proxy of the decoder RNN hidden state)—this sometimes translates as a 'real' intuitive alignement, as in the dot and general models, but not always, as with the concat alignement or the FFNN in Bahdanau et al. (2015). It therefore improves the transformation of long sequences, as well as introduces an intuitive way to look at the alignement between hidden representation and output (Luong et al. 2015a).

Comparison Two points are interesting to note:

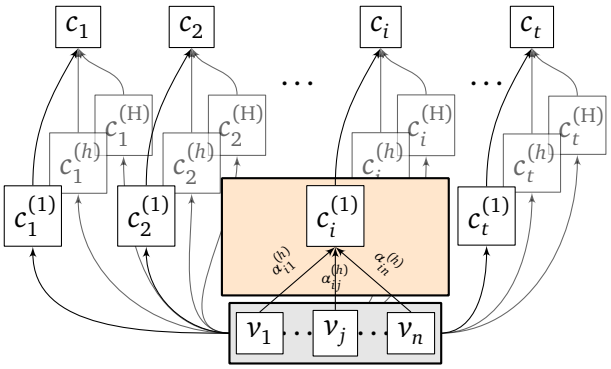
- Computation paths differ: Bahdanau's decoder runs the attention on the encoder output, before providing it to its RNN: $s_{i-1} \rightarrow a_i \rightarrow c_i \rightarrow s_i$, where Luong's decoder runs its RNN first, and uses its output to compute the attention: $s_i \rightarrow a_i \rightarrow c_i \rightarrow \tilde{s}_i$.
- The decoder hidden states considered are not the same: the decoders use the context, plus, for Bahdanau's decoder, the **previous** hidden state and current target, and for Luong's model, the **current** hidden state.

2.3.3 Transformers, Attention without Recurrence

Recurrent networks are computationnally intensive and hard to parallelize because of their sequential nature. To use attention while removing sequentiality, Vaswani et al. (2017) introduced a new model: the Transformer. This time, the model does not contain any recurrent layer, but only attention mechanisms organised in an encoder-decoder setup,[23] via a new attentional unit: the multi-head attention.[24]

2.3.3.1 Multi-head attention

In multi-head attention units, there is no recurrent module. All input words are seen and studied at the same time. 'Multi-head' refers to this parallel mechanism, where several alignement models (called heads) are computed at the same time for the same inputs; the hope of this parallel computation is that each different alignement model, initialised randomly, learns to focus on a different aspect of item relations in a sequence.



1. **Total context of item i**: $c_i = \left[c_i^{(0)} ... c_i^{(H)}\right] W_c$
2. **Target item i context at head h**: $c_i^{(h)} = \sum_{j=1}^{n} \alpha_{ij}^{(h)} (W_v v_j)$
3. **Local context weight**: $\alpha_{ij}^{(h)} = \sigma_j\left(a^{(h)}(v_j, t_i)\right)$

Figure 2.9: Schematic representation of a multi-head attention unit.

In more detail (Figure 2.9), each input item v_j is aligned with a target t_i[25] to compute normalized scores $\alpha_{ij}^{(h)}$, which represent how well the input matches the target (4). Said scores

[23] As well as FFNN as transition between layers, but we will not detail them here.

[24] Recurrent models see each word of a sentence in a sequence, and therefore know which word is close to which one. Transformers see all words of the input at the same time without information about their respective positions. To mitigate this, their first unit, before the encoder and the decoder, is a **positional encoding layer**, which adds word order information to the input embeddings, but will not be detailed here. For more information, see Vaswani et al. (2017).

[25] What the target can be is detailed in the next paragraph.

are then used to compute the weighted sum $c_i^{(h)}$ of all inputs v_j (3), which is, in a sense, the Transformers' item 'context'. All the contexts $c_i^{(h)}$ are actually computed at the same time in one step, using matrix computations (where, in recurrent models, items were looked at sequentially). Since heads are parallel attentional models (each hopefully learning to specialize on a different aspect), all these computations are done at the same time, in parallel. The concatenation of all head contexts $c_i^{(h)}$ for a given item i, through a linear layer (W_c), constitutes the final context c_i of the given element i. Here, the alignement model is similar to the general attention (Luong et al. 2015a).

The superscript $^{(h)}$ indicates that these computations are done for a given **head** h.

Variations

1. **Multi-head self-attention** If a multi-head attention layer is used to compute the alignement of a sequence with itself ($t = v$), it is called a multi-head self-attention layer. These layers highlight which items of a given sequence are most relevant to other items of the same sequence, and therefore allow the model to learn about items relations in a single sequence. For example, in 'The students ate their cake.', 'their' has a relation to 'The students'.
2. **Multi-head masked self-attention** Sometimes, we want our self-attention alignement models to be constrained, and to only link an item in a sequence with previous items in the same sequence, and not following items (which mimicks recurrent attention, where the decoder at time t only has access to its previous predictions). To do so, 1) the alignement weights $\alpha_{ij}^{(h)}$ are contrained to be null for i>j (using what is called a mask), and 2) the input sequence is aligned not strictly with itself, but with itself **right-shifted**: a <BOS> token is prepended to the sequence (t = [<BOS>, v]).

2.3.3.2 Full Transformer model

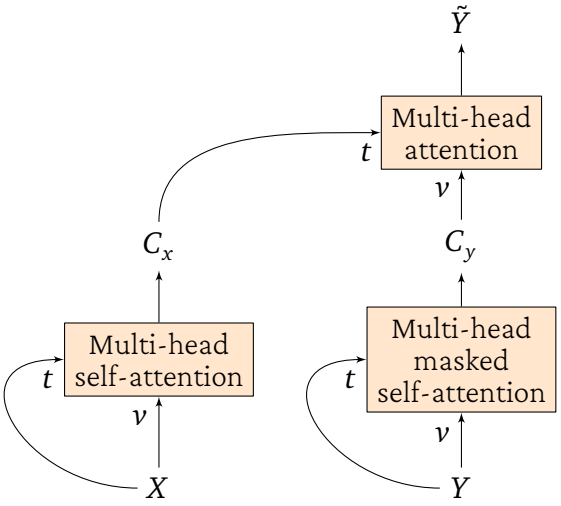


Figure 2.10: Schematic representation of a Transformer.
(v and t on arrows indicate if the parameter is used as value or target in the previous equations).

The Transformer, in its encoder, contains a multi-head self-attention unit, which computes

a representation C_X of the input sequence X as the relation of its items x_i. Then, the Transformer decoder uses a multi-head masked self-attention unit to compute a representation C_Y of the gold sequence Y as the relation of its items y_i with their predecessors in the sequence y_{i-1}. This allows us to show the full gold at once to the decoder (which allows parallelization) while not allowing the decoder first unit to link an item to its successors in the sequence: predicting the next item if you already know it is not really a challenge, nor can it generalize to unknown sentences. The decoder finally uses a multi-head attention to align the encoded input representation C_X with the gold masked representation C_Y to link each input word with a 'sub-sequence' of the output, and infer its prediction $\tilde{Y}$.

Teacher forcing was not always used for Transformers, since the original paper explains that the model is 'auto-regressive at each step'. However, most current implementations use teacher forcing for decoder training, as it increases computational speed several folds.

2.3.4 Attention?

Both architecture types (Seq2seq vs Transformers) use the same mechanism (attention), but conceptually explained differently. Bahdanau et al. (2015) and Luong et al. (2015a) refer to attention as an alignement model (concept which comes from statistical translation),[26] as seen earlier: the current 'source' is aligned with its 'target' to determine alignement weights (using a dot product), then used to ponderate the 'source' impact on the output. Vaswani et al. (2017) refer to attention as a system of queries, keys, and values (concept which comes from information retrieval systems): the queries are mapped against the keys (using a dot product), and the result is used to weight the most relevant values. From these descriptions and when comparing Figure 2.8 with Figure 2.9 (most notably the arrows going from the grey input to the context), we can see plainly that the queries and values match the 'source,' and the keys match the 'target': we map the 'source' against its 'target' to determine the more relevant parts of the 'source'.

2.3.5 Decoding

Training When training RNN based encoder-decoders, we can either teach them to predict a target token from the previous prediction (**student forcing**), or from the actual previous gold token (**teacher forcing**).[27] When decoding in the former case,[28] being wrong on the first token likely means being wrong on the second one, and then on the following ones, as each token is predicted from a previous one, and error can therefore propagate; using student forcing teaches the model to mitigate its own mistakes, whereas a model trained using teacher forcing can suffer from exposure bias: having only learned to predict from the gold, it has problem inferring from its own predictions. However, teacher forcing also prevents the model from drifting too far away at training time, since the decoder learns to predict from the correct token, and this helps faster convergence. Where the choice does not change a thing computationnally speaking for recurrent decoders, which will predict token one after the other, it is very different for Transformers, which can predict all target tokens at once. Transformers are therefore usually trained using teacher forcing.

Inference A model trained using teacher forcing will not have access to the whole gold target during inference, which is solved using a technique called **greedy search**. The decoder, start-

[26] This concept is also similar to DNA sequence alignement models in biology.
[27] The first target token to predict from is always a token indicating 'start of sentence here'.
[28] This is also called auto-regressive decoding.

ing from the 'start of sentence' token, predicts the most plausible token. From the formed sequence, it predicts the next token, and so forth. It is extremely similar to student forcing, except that model weights are kept fixed (since we are inferring). However, it can sometimes fail if the best token at one point is then followed only by very unlikely tokens, when considering the next best choice would have been followed by good tokens, giving an overall better prediction in the second choice. To mitigate this, greedy search can also be ran while keeping not the best, but the n-best predictions at each time step, and chosing the overall best path at the end. This is called **beam search**. It is considerably more computationally intensive, but can provide better results.

2.3.6 Autoencoders

Autoencoders are a special case of encoder-decoders, which differ from them only on the task they were trained on. The first autoencoders (Kramer 1991),[29] called autoassociative networks, were designed to solve how to extract relevant low-dimensional features (interesting properties which summarize the data correctly) from highly dimensional data in an unsupervised manner (without being told what to look for). An auto-encoder is trained on 'self-to-self' prediction: it learns to map input data to a lower-dimension feature space using the encoder, then reconstruct the input from the feature space representation with the decoder. Several constraints have then been applied to the initial concept to prevent the network from learning the identity function: forcing network **sparsity** by shutting down some of its neurons during learning, adding noise to the input data and training on **denoising** it,[30] and others. The encoders can then be used as such to generate lower dimensional representation of new data. This is one end of the pretraining/fine-tuning spectrum mentioned in Section 2.1.2.

2.4 Conclusion

Throughout this section, we successively studied how neural networks are conceived and translated into equations, before studying some neural networks applied to text, from recurrent units (SRNN, LSTM, GRU) to encoder-decoder architectures (recurrent or Transformer based). These are not the only useful artificial neural networks when studying texts, but those which will be used throughout this manuscript.[31] However, though this section might have given the reader the feeling that neural networks are all pure mathematics, this is actually only true in theory. In practice, machine learning models need a lot of adjustment, as we will see in the next section.

[29] Autoencoders predate encoder-decoders by two real world decades, or two centuries in machine learning time.

[30] Masked Language Models (MLMs) are a special case of denoising, where some input tokens are masked and the network must learn to predict the original sentence, as in BERT models (Devlin et al. 2019).

[31] Preliminary experiments have also been conducted with convolutional or variational encoder-decoders, with little success, and they will not be detailed in this manuscript.

3 Machine learning is an experimental science

> What is the difference between
> theory and practice?
> –
> In theory, they are the same.
>
> My grandfather

The field of applied machine learning is as much an experimental science as a purely theoretical one, and it needs to be considered as such. Many parameters can be adjusted at each step, whose variations can give drasticaly different final results. In this section, we explore some of these parameters, from data management to model design and training, then focus on an important aspect of experimental sciences: reproducibility. Lastly, we introduce some interpretability concepts, a recent but nonetheless vital aspect of machine learning.

3.1 Data

3.1.1 Obtaining a dataset

3.1.1.1 Collection

After having defined the task, the next step is to gather data, either by manually producing it for the task at hand (which is resource expensive), or by reusing existing datasets. The latter present a few challenges, as a balance must be found between **data size**, big enough to train neural networks, **cleanliness**, to allow for good precision, as well as **variety and representativeness**, to allow for generalisation (see e.g. Barbosa and Feng (2010) and Soni and Roberts (2020) for a description of the challenges of using datasets of different sizes, noise levels, and relevance, for domain-specific tasks, Schäfer (2016) for a discussion of the biases of crawled datasets, and Kreutzer et al. (2022) for an audit of widely used crawled sets). Bender and Friedman (2018) and Rogers et al. (2021) provide frameworks to constitute and think about datasets.

3.1.1.2 Augmentation

When the task of interest does not have enough annotated data in the language at hand, it can be useful to use data augmentation techniques (especially when resources to create a relevant data set from scratch are not available). This can be done by changing the datasets, leveraging information from other more available sources, or by creating artificial synthetic data. More concrete examples of data augmentation will be detailed applied to machine translation, in Section 4.2.1.

3.1.1.3 Sampling

As seen above in data collection, data sets should be representative of the task at hand. However, if datasets (even reflecting reality) are too imbalanced, models can sometimes be stuck learning only majority examples, even noisy ones, to the risk of overwhelming the minority class (Japkowicz 2000; Kubat and Matwin 1997; Laurikkala 2001; Maragoudakis et al. 2006).[1]

Data might then have to be edited depending on the task objective, and whether it is more important to get good precision (percentage of predictions which are correct) or recall (percentage of target which was correctly predicted). Two principal strategies can be applied to try to balance a dataset: **undersampling** means reducing the total size of the data by removing samples from the 'majority class,' whereas **oversampling** repeats samples from the minority class and increase therefore the total size of the data – undersampling tends to performs better than oversampling (Chawla et al. 2002). Their effects have been studied over a wide range of topics (Jamil et al. 2017; Li and Scarton 2020; Lichouri and Abbas 2020; Washio and Kato 2018; Zhu and Hovy 2007).

For more complex tasks, however, oversampling might lead to overfitting the specific examples we repeat. In those cases, it might actually be better to change the architecture or loss chosen for the problem to focus on better management of small classes.

3.1.2 Preparing data for training

Once data has been obtained, it must be prepared for training.

Cleaning Data should first be cleaned before use if needed, in order to remove as much of the noise as possible without losing performance, such as items that are obviously incorrect (e.g. a wrong character set for the given language, such as Chinese ideograms in an Alsatian text, a wrong label, etc.), or plausibly incorrect, being too different from the others (bad ratio of character to word length, sentences considerably longer or shorter than the average, unusual frequency

[1] This can also apply, not directly to the data points, but to their metadata: for example, Geva et al. (2019) found that their models do not generalize well to data annotated by new annotators, who did not contribute to the training set - they effectively learned on the 'majority annotator class'.

of target labels, etc.) (Koehn et al. 2018).[2] Several automatic tools have also been developped and published for data cleaning and filtering (Aulamo et al. 2020; Bane and Zaretskaya 2021; Laippala et al. 2020; Muthuraman et al. 2021; Ramírez-Sánchez et al. 2020).

Tokenization Cleaned data is then tokenized at the chosen level: segmenting a running input into 'studiable' and 'processable' units (**tokens**) is called **tokenization**. In NLP, several units are possible at the sentence level, from whole word to characters. Tokenizing a sentence into orthographic **words** makes intuitive sense, but it is less trivial than it seems (as seen in subsection **??**), and it creates a huge list of unique units, the **vocabulary**. Tokenizing into **characters** reduces the vocabulary size and is more trivial, but it increases the length of the studied sequences, which can make sequence modeling harder by introducing too long-term relations.[3] A good intermediate seems to be **subword** tokenization, rule based or statistically motivated (as in the Morfessor family of tools). Morphologically motivated subwords allow us to represent words out of sensical units (for example, prefix + root + suffix), but require prior knowledge of language to build the associated rule-based tokenization models.[4] Purely statistical segmentation has been introduced to solve this problem, using a technique called byte-pair-encoding (**BPE**): the vocabulary is initialized with all the unique characters available, then the text is parsed to count frequent vocabulary n-grams (which take into account word boundaries). The most frequent character n-grams are considered as new tokens, and added to the vocabulary. This operation is then repeated iteratively, until a number of operations or vocabulary size (Sennrich et al. 2016b). BPE has become the segmentation of choice in a lot of tasks, thanks to its performance and easiness to setup. Comparing different tokenization levels (character, morphologically based subword, and BPE based subword) for several natural language processing tasks highlighted that character level tokenization help to better learn morphology, BPE semantics, and morphologically-based subwords syntax (Durrani et al. 2019), which interestingly correlates with the previous comments about subword sentence segmentation in linguistics being enough to analyse language relevantly (see also (Stahlberg 2020):9.4). The vocabulary size constitutes a parameter in itself in settings where it is fixed *a priori*.

Data split Data split is not usually seen as an hyperparameter, but dividing our total set into training, validation and testing sets can have a huge impact on the overall accuracy of the model. If the splits are not balanced similarly, and if, for example, a minority class is completely absent from the testing set, the model is likely to report better performance than what is actually true, since it will never be evaluated on the harder minority class. Two strategies exist: first, being careful about the data division when creating train/val/test sets, or second, running the experiments several times with several different data splits. (This second option is more robust, but tends to be unfeasible in practice because of model training times.) The relative size of our subsets can also affect performance — more training data might seem better, but using too little testing data might prevent from seing if the model fails to generalise. These points are discussed by Goot (2021) and Søgaard et al. (2021). The usual data splits for average corpora in common

[2] See also MOSES cleaning scripts `github.com/moses-smt/mosesdecoder/blob/master/scripts/training/clean-corpus-n.perl` and `github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/normalize-punctuation.perl`, or this data cleaning repository `github.com/jfilter/clean-text`.

[3] See Subsection 2.2 for more details.

[4] They can also have trouble segmenting words containing discontinuous morphemes (morphemes into which other morphemes have been inserted), such as *bumili* 'bought' in Tagalog, made of *bili* 'buy' interrupted by the infix *-um-* which changes word meaning depending on grammatical context.

tasks (e.g. POS tagging or semantic tree labelling) are usually around 80%/10%/10% for training/-validation/testing; for very big corpora (as in machine translation), validation and test set are no longer a matter of percentage but of absolute size (5,000 sentences constitute a big enough validation set); for very small corpora, a bigger training set is sometimes kept at the expense of the validation set.

Batching As mentioned in Section 2.1.2, it is possible to provide input sentences as mini-batches (several at a time) when training. Using batches or mini-batches[5] increases performance and computation speed, by going for fewer but larger operations (not n small operations, but one big operation on a n-size matrix). These larger computations can then be run even faster by taking advantage of parallelisation. However, working on too many examples at the same time can decrease precision, as the model backpropagates and optimizes for an error averaged over n elements, instead of updating for the error of one or a few examples. In NLP, it is rare to find a data set where all examples are the same size, since we look at text (sentences, contrarily to images, do not have 'standard shapes'). This poses a problem for batching: creating a table of examples to be recursively parsed together requires all sentences to be the same size, as the model will look at all first words at the same time, then all second words, and so forth. Two solutions exist to this problem. The first one is called **bucketing**: mini-batches are made by grouping together chosen examples which actually are the same size. The second is called **padding**: too short sequences are completed with a repeated 'fake item,' to make them as long as the longest sequence of the mini-batch. However, both padding and bucketing present their own limitations: padding, by adding fake tokens, adds useless model computations on said tokens, while bucketing, by grouping same-size sentence, might affect model stability for the rare length classes, which will contain less items than the others (see Morishita et al. (2017) for a study of mini-batches creation strategies, though most papers use bucketing or random shuffling with padding).[6]

3.2 Experiment design

3.2.1 Architecture choice

Architecture choice aims at choosing the best architecture for the task at hand, either from pre-existing ones (e.g the different encoder-decoder flavors we saw in the last section) or by designing a new one (specific to the task).

3.2.1.1 Model size

Once the general architecture is chosen, its parameters must be chosen (number of layers and layer sizes, type of units, etc). The total number of model weights and biases must fit the size of the data and task complexity: in theory, too small a model will not be able to approximate the complexity of the data and will not learn, and too big a model could risk becoming too good

[5] The difference is explained in the side-note of Section 2.1.2

[6] During training, data is usually shuffled at each epoch to help preventing overfitting - however, methods such as bucketing force specific samples to always be seen together (especially for small classes), therefore introducing non-randomness and possibly hurting generalization.

an approximator of its training set patterns, and overfit (see Section 2.1.3.1). However, since large models overfitting is not systematic, as some learn to generalize on complex enough tasks (Caruana et al. 2001; Zhong et al. 2021), big models can also become synonymous with better models.

In language modeling, this created an arms race to build the biggest language models at strong ecological and financial costs (themselves creating 'barriers to entry' for less funded labs or teams, opportunity costs, as well as accountability issues due to the excessive complexity of models) (Bender et al. 2021), when smaller, but smarter models can reach similar performance at smaller costs (Schick and Schütze 2021).

3.2.2 Model instanciation

3.2.2.1 Weights initialization

Models weights can be initialized before training on the task of interest. This initialization can be mathematical (random or using a specific function), data-based, or model-based.

Data-based Initializing a model using a dataset is called **pretraining** the model. A special case of pretraining, called **transfer-learning**, uses data from a high-resourced related task to initialise the model, and hopes information from the high-resourced task can be used by the low-resourced task (Gu et al. 2018; Ruder et al. 2019). If the model is not trained after and used as such (for example, if there is no training data for the low-resourced task), this is called **zero-shot transfer** (Lauscher et al. 2020).

Model-based It is also possible to use parts of pre-trained existing models as components for our model of choice. The most common initialisation of the type is models trained on a language modeling task,[7] which enrich sentence representation, though they can negatively impact the performance of models after training (Wang et al. 2019). Well known recent neural models include:

- **BERT** masked encoder models: Bidirectional Encoder Representations from Transformers, having learned to predict the value of tokens masked in inputs (Devlin et al. 2019),
- **GPT** decoder models: Generative Pre-trained Transformer, predicting the next token in a sequence (Brown et al. 2020),
- **BART** denoising auto-encoder models: Bidirectional and Auto-Regressive Transformer, learning a combination of both tasks with added noise (Lewis et al. 2019),
- **T5** encoder-decoder models: Text-to-Text Transfer Transformer, learning language modeling through prompting[8] for a combination of tasks (Raffel et al. 2020).

3.2.2.2 Embeddings

The input's tokens must be transformed into numbers to be provided to our models. This can be done using an **embedding layer**. In its simplest form, this layer is a lookup table (linking tokens with indices). Since the introduction of language modeling (Bengio et al. 2003), embedding layers tend to link tokens to vectors in a continuous space, learned unsupervisedly by models con-

[7] Language modeling is a task where a model is trained to reconstruct a full sentence from a sentence with missing information, by learning to predict either the content of masked sequences in a sentence, or the end of a sentence given its start.

[8] Prompting a model means providing it with a description in natural language of the task to do.

strained to map items occuring in similar contexts to mathematically close representations (language models for word items), thus carrying more information than just an alphabetical index (Mikolov et al. 2013a). These vectors are called **embeddings**, and can be initialised randomly or from a learned coherent representation (coming from a language model for example), then fine-tuned unsupervisedly during training or fixed (Baroni et al. 2014; Levy et al. 2015; Li and Specia 2019a). **Word embeddings** are interesting, because they can carry semantic information; however, they introduce a challenge at test time, as models will be unable to work with words which have never been seen by the embedding model – this is called the **out-of-vocabulary** words problem, for which many strategies have been developed (Daumé III and Jagarlamudi 2011; Jean et al. 2015; Yang et al. 2020a; Yazgan and Saraclar 2004). **Character embeddings** do not suffer from this problem for languages where the total character set is limited, such as French or English, but as we saw earlier, character tokenization is not good for long-term relations modeling. **Subword embeddings** is a good intermediary, as it almost never suffers from the out-of-vocabulary words problem, while still having the potential to carry semantic information (Durrani et al. 2019). Of course, embedding choice depends on tokenization, but embedding size and initialisation are hyperparameters in themselves.[9]

3.2.3 Training

Many hyperparameters also have to be adjusted when training, among which the learning rate (speed at which the optimization occurs) and its possible scheduling (changing the learning rate ratio during training to make bigger steps at the beginning of optimization and smaller afterwards), the optimizer chosen (from the robust SGD to faster Adam (Kingma and Ba 2015),[10] see Ruder (2016) for an overview), the batch size and batching technique (from bucketing to padding and everything in between), the maximum number of epochs during which the model is allowed to run, the convergence metrics which determine when the model has reached its optimum, and of course the loss computation, which determines what is important in training a model.

3.2.4 Mitigating variation

It is impossible, in terms of time and computing power, to train on all subsets of hyperparameters. In order to mitigate variation as much as possible, a user can choose an informed subset, and experiment around it, doing what is called an **hyperparameter search**: this implies training several models to study their performance with respect to the chosen parameters (Liu and Wang 2021). It must take into account the fact that some of the forementioned parameters are linked (batch size should be changed along with learning rate, for example). Another option is to use **ensembling**, and to train a varied number of models, and merge their results (by averaging them for example) (Gautam et al. 2021).

[9] Other common embeddings are **sentence embeddings**, used to work at the document level, but out of scope for this document.

[10] Several recent papers have shown that adaptative optimizers (such as Adam, Adagrad, etc), though considerably faster than the simpler SGD, also tend to suffer from worse generalisation performance (Wilson et al. 2017; Zhou et al. 2020).

3.2.5 Evaluation

Baseline In order to evaluate the results of a given model, they must be compared to a good baseline. The usual method in machine learning is to find previous work, which used the same data set and same architecture as baseline. However, this is complicated by the only occasional occurences of completely detailed experimental setups in exising machine learning papers,[11] as well as the difficulty of discriminating which parts of the results difference come from minimal design variations versus new techniques introduced. That is, when previous papers have worked on the same available datasets and task (Marie et al. 2021). If no one has worked on the same subtask before, it can be interesting to compare the results of neural methods with previous non neural methods, such as, in the case of translation, rule-based or statistical machine translation. Another possible baseline, though more expensive, is the human one.

Scoring Metrics must be chosen carefully to evaluate our models on the most important aspects of the task. Some classical metrics are precision (percentage of predictions which are correct), recall (percentage of target which was correctly predicted), a combination of both using a score, or the accuracy of a task-specific metric. However, desigining automatic task metrics of good quality can be a challenge, especially for more complex task such as translation, summarization, and generation, with many papers pointing their limits (Mathur et al. 2020; Reiter 2018; Reiter and Belz 2009; Sulem et al. 2018; Weber et al. 2021).

3.3 Reproducibility

Reproducible research provides enough information to allow other people to replicate results (or try to), from papers in open access, documented and open source code, used datasets under open licenses, ideally with data splits, model checkpoints, and so forth (see for example the European Commission report on scientific data (Wood et al. 2010) or the Open Linguistics Working group recommandations (McCrae et al. 2016)). Some initiatives to help generalizing this approach have emerged, such as 'Papers with code',[12] and providing code and data starts to be integrated into paper evaluation for the main NLP conferences.

However, despite those efforts, machine learning tends to be inherently unreproducible: most machine learning libraries will not give the same results on the exact same dataset, task, and model architectures. To quote the PyTorch documentation, 'Completely reproducible results are not guaranteed across PyTorch releases, individual commits, or different platforms. Furthermore, results may not be reproducible between CPU and GPU executions, even when using identical seeds'.[13] This is, in a lot of cases, due to the use of nondeterministic algorithms (notably when parallelizing with Cuda) to increase computation speed.

Sennrich et al. (2016b) reported difference up to a full BLEU[14] point between different in-

[11] However, this is sometimes mitigated by providing the codebase used to run experiments.
[12] `http://paperswithcode.com`
[13] See `http://pytorch.org/docs/stable/notes/randomness.html`
[14] See Section 4.1.1 for a definition

stances of the same models trained on the same data with the same number of epochs.[15] Therefore, apart from some steps (detailed in the different libraries documentations) that can be taken to enforce the use of deterministic algorithms and set fixed seeds for random operations, the best attitude towards machine learning is most likely to **treat it like an experimental science**, and as such, to run experiments several times, with different seeds, or on different machines, to ensure the results are statistically significant. This is especially important when comparing architectures, as a specific run for a given architecture, on a single seed and data split, could happen to outperform an actually better architecture, which was slightly less good for this specific situation. Of course, this might be hard to do for the bigger NLP tasks (translation, language modeling), but in this case, it might be better to try to design more interpretable models, to at least add a level of human understanding of the model choices (Bender et al. 2021).

3.4 Interpretability

Machine learning models behave as black boxes, and it is close to impossible to understand what is happening under the hood. They usually are evaluated on specific scores for a given task and dataset, which constitute a very reduced interpretation of what is actually happening (as shown by Elsner et al. (2019) for morphological modeling capabilites of seq2seq models), and do not explain artefacts and border cases. Besides, machine learning models are now being used on life-impacting issues, such as in the legal or medical domain. There is more than a theoretical need to understand how they work, notably to provide accountability (Bender et al. 2021).

Interpretability, studying and interpreting the inner workings of such models, is a very recent field in NLP, as the first workshop dedicated to the topic occured in 2018, colocated with EMNLP.[16] As such, a lot of basic definitions are still debated. Madsen et al. (2021) wrote a survey on interpretability techniques applied to NLP, and introduced the term *post-hoc interpretability*, for techniques applied a posteriori to models already trained – this is also what we will focus on. We will first introduce some necessary distinctions, then present a panel of model interpretability techniques, either external (focusing on inputs and outputs) or internal (studying the inner components of models).

3.4.1 Useful distinctions

Interpretability vs explainability Interpretability is, according to Doshi-Velez and Kim (2017), the "ability to explain or to present in understandable terms to a human" what is happening in a model. Li et al. (2021) define interpretability as when "there exists a trustworthy interpretation algorithm [such that] (1) the rationale behind the model is fully revealed by the algorithm; and (2) the revealed rationale is totally understandable by humans, or fully overlaps with human understandings". We define explainability as, on the other hand, knowing what is happening in the model.[17] We could see interpretability as an understanding plausible in human terms, where explainaibility is a full mathematical understanding of the reasoning. At the

[15] The only variation between their models was the epoch at which the embedding layer was fixed (last 4 epochs before fine tuning) and the gradient clipping rate during fine-tuning (1.0 or 5.0).

[16] `http://blackboxnlp.github.io/2018`

[17] As mentioned above, there is no consensus on the two terms - for a overview, see Fan et al. (2021).

moment, a neural machine translation model is at most interpretable, where a rule based translation model is explainable, as it has been designed as such from the ground up. We focus here on interpretability, not explainability.

Faithful vs plausible Jacovi and Goldberg (2020) insist on the difference between faithful and plausible interpretations of our models. When designing interpretability datasets or probes, we must be careful to avoid 'plausible' interpretations writing pretty stories around what we observe, and rather do our best to make sure our explanations are faithful to what is actually happening. It can be easy to be biased in task design, to try to fit what we expect to interpret.

3.4.2 External analysis of models

External interpretability is, in a sense, a 'blackbox', as it only uses model inputs and outputs to infer what is happening inside. For example, it is possible to design edge case inputs to see how the model reacts; it is also possible to design further experiments to 'plug' in the model's representations, to better understand what they could contain.

Creating artificial test data Artificial test data are designed to target specific model capabilities, or help analyse model errors, and predictions made on such data can help learn more on the edge cases of modeling. For example, in their tutorial on interpretability applied to NLP models, Belinkov et al. (2020) introduce techniques for dataset design focused on linguistic testing, such as tests sets designed to overcontain minority cases and sentences to target specific properties to look for. Similarly, Kuhnle and Copestake (2018) automatically generate artificial data for visual question answering, to better account for sentences complexity level (syntactic richness), reflecting reality more interestingly than turked[18] datasets. These datasets focus on aspects of linguistic complexity we want to study in our models. A specific case of artificial test data are **adversarial examples**, inputs that cause the model to output something completely wrong unexpectedly, either false (mis-classification) or problematic (slurs or private information production in text generation). Wallace et al. (2019), for example, design an iterative replacement search over tokens to find adversarial sequences which trigger specific predictions for different tasks (classification, text generation, sentiment analysis, etc.). These tell us more about the edge cases of our models, and how much it has overfit its initial data.

Using probing downstream tasks Probing tasks are designed to analyse if model representations encode properties of interest. Belinkov et al. (2020) use a classifier to predict a property of interest from a model representations of the input; the accuracy of the classifier is seen as a proxy for the quality of the representation with respect to the studied property. Voita and Titov (2020) propose another proxy — not just probing tasks accuracy scores, but the "cost" of getting this accuracy: the minimum description length needed for the probing task, which indicates how accessible this property is in the representation (the longer the description needed, the less accessible the property). Probing studies a specific aspect of language, as when Tang et al. (2020) use classifiers on model internal components, to probe character level hidden states on

[18] 'Turked datasets' refer to datasets created using crowdworkers on Amazon mechanical turk or equivalent systems, whose use presents strong ethical issues (Fort et al. 2011).

different morphological tasks and determine at which layer which morphological information is learnt by such a model. Therefore, many papers combine different probing levels to grasp a finer picture of learning: Conneau et al. (2018) for sequence to sequence models, Raganato and Tiedemann (2018) for multilingual Transformers's encoders and attentions, or Jawahar et al. (2019) for language models.

3.4.3 Internal analysis of models

Model components visualisation Internal models representations (such as the contexts for encoder decoder) can be visualised using dimension reduction techniques, which map this multi-dimensional input to a lower space representation. For example, PCA (Primary Component Analysis) applies a linear transformation to the data, in order to fit it to a new coordinate base, chosen to maximize the variance of the data for all successive components (Pearson 1901). t-SNE (t-distributed Stochastic Neighbor Embedding) defines two probability distributions, one on the original data points (where the closer the pair, the higher their occurence probability) and one defined on the chosen lower dimension output space (in the same way). The two probability distributions are then aligned together by minimizing the Kullback-Leibler divergence, to fit original data points to the target space (Maaten and Hinton 2008). This method is stochastic, and can generate results which vary from one run to the next. Madsen et al. (2021) provide an overview of such visualisation techniques for NLP problems.

Input importance visualisation It is also possible to litterally visualise input importance with respect to the prediction or inner components of the model. Some methods use saliency maps, retrieving neuron-wise signal contribution of inputs to results (Harbecke et al. 2018), look for inputs which maximize the activation of a neuron of interest (Poerner et al. 2018) or study the relevance of inputs for different internal components (hidden states, embeddings) layer wise, relevance then displayed using matrices (Ding et al. 2017).

Interaction Some authors change inner model weights while looking for properties in said models, such as Giulianelli et al. (2018), who train a diagnostic classifier to predict verb number and number agreement from corresponding LSTM states of a language model, then backpropagate the classifier gradients to change the LSTM state - it does not improve the language modeling accuracy, but the scores on accordance tests. Wiegreffe and Pinter (2019) train models to find adversarial attention weights, to determine the limits of using attention as explanation. Bau et al. (2018) analyse which neurons are the most relevant to their task and its subtasks. This is not an avenue we will elaborate more on.

3.5 Conclusion

In this section, we looked at the experimental aspects of machine learning, from data collection, preprocessing and tokenization to training parameters and evaluation choice, not forgetting of course the many existing models and model parameters. We will therefore need to take hyperparameter optimization into account when designing and running experiments. We then highlighted how unexpectedly hard it is to reproduce machine learning results exactly, introducing

the need for statistical significance if possible (much as in biology, for example), or else interpretability: we therefore looked at a number of interpretability tools, some of which will be used to better understand our results.

4 Inspiration from low-resource machine translation

> What kind of language is this?
> I can't hear a word you're saying
> Tell me how you are singing
> In the sun
>
> Bush (2005)

Now that we both studied the theory behind our units of interest, as well as the attention points to keep in mind when training machine learning models, we look at our inspiration for our architectures and tools. As our tasks of interest are sequence-to-sequence problems with very small datasets (cognate prediction and proto-form reconstruction tasks), they seem theoretically very similar to the low-resource machine translation task. We will therefore describe here the basics of machine translation, then the specifics of low-resource machine translation, most notably on how to improve performance when data is scarce.

4.1 What is low-resource machine translation?

4.1.1 Machine translation

Machine translation (MT) uses computer models to translate text from a language to another. When working with written data, it is usually considered to be a **many-to-many sequence-to-sequence** problem, as we go from one sequence (a sentence in a source language) to another sequence (its translation in the target language), and those two sequences can have different number of tokens or different ordering of the tokens. For example, if the unit we chose for our task is the word, English *'Cats eat mice.'* gives French *'Les chats mangent des souris.'*, and we go from three units to five.

Datasets In MT, datasets of interest are called **parallel datasets**, as they contain parallel sentences in usually two languages, which means they store the correspondences between each sentence and its translations. Available datasets are usually either big and noisy, such as CommonCrawl, data crawled from the web, or cleaner but smaller and/or specialised, such as WMT

datasets, manually curated for MT competitions, or EuroParl, multilingual dataset of European parliament deliberations.

Approaches Two approaches remain in machine translation.[1] Historically, machine translation was performed using statistical approaches, based on computing glossaries of correspondence between source and target, coupled with reordering models for output (Brown et al. 1988). The reference tool for **statistical machine translation** (SMT) is called MOSES (Koehn et al. 2007). It works in two steps, training and fine-tuning. During training, the bilingual parallel data are tokenized and aligned using an external statistical alignement tool, then used to learn 1) an n-gram language model of the target language; 2) a correspondence table between source and target tokens; and 3) a reordering model to manage token order. During fine-tuning, the respective weights of all these models are adjusted, using a development dataset (Mikolov et al. 2013b). **Neural machine translation** (NMT) emerged with the success of neural encoder-decoder architectures specifically developed for this task, either recurrent with attention (Bahdanau et al. 2015; Luong et al. 2015a) or fully attentional such as the Transformers (Vaswani et al. 2017).[2] These models have been implemented in several toolkits, among which `fairseq` (Ott et al. 2019), which we use. Overall, when **comparing**, SMT models have the advantage of being more understandable, as they are made of specific, separate components (alignment model, language model, decoder) which can be studied on their own; this information is implicitly learned by NMT models. However, NMT models can learn to go from several languages to several others, which is not possible using SMT. This multilinguality can be managed using a panel of techniques, from sharing a single encoder and a single decoder across all languages (Johnson et al. 2017; Vergés Boncompte and R. Costa-jussà 2020) to using one component (encoder/decoder/attentional mechanism) for each language (Luong et al. 2015a), and any combination in between, such as sharing encoders only (Li et al. 2020a), components across related languages, or sharing attention only across all (Firat et al. 2016a). We will see their respective performances in the next section.

Evaluation For translation, model results were historically manually evaluated, which is time consuming and expensive. Several automatic metrics were therefore introduced. **BLEU** is a 'translation closeness metric' inspired by the word error rate used in speech processing (Papineni et al. 2002). It uses modified n-gram precision scores on blocks of text, averaged using the equivalent of a geometric mean to compute a precision score. Post (2018) introduced a framework to communicate BLEU scores more homogeneously and accurately. Because of its apparent good correlation with human judgment for MT tasks (but not generation tasks) (Reiter 2018), it has become the "de facto standard for evaluating research hypothesis" (Mathur et al. 2020). However, the authors of this second paper, evaluating MT evaluation metrics, note that BLEU correlation to human judgement and other metrics decreases considerably when removing outliers MT systems, over or underperforming, from the comparison: as systems get more complex, small differences in BLEU might have less meaning and value than could previously be thought. **ROUGE** was introduced as a BLEU alternative for summarisation, and uses either the overlap of n-grams between prediction and reference or the longest common sub-sequence as recall metrics (Lin 2004). BLEU and ROUGE are quite complementary, as BLEU measures precision (e.g how many predicted terms are correct, and were originally present in the reference sentences) and ROUGE measures recall (e.g how many words in the reference sentences can be found in the predictions).

[1] Rule-based approaches have also been used, but they fail to scale and are rarely kept today.

[2] Both model types have been described in detail in Section 2.3.

METEOR combines the two, using the harmonic mean of 1-gram precisions and recalls (Banerjee and Lavie 2005) to get an overall score. These are not the only metrics existing, but the most commonly used (other metrics attempt, for example, to remove the reliance on gold data for evaluation (Zhao et al. 2020), or to focus on model robustness to noise (Niu et al. 2020)). Lastly, evaluators made of neural networks have also been introduced, such as **COMET**, combining a pretrained encoder to extract information about hypothesis, source and reference, and training a minimal neural network on minimizing either 1) the mean squared error between sentence representation and quality assessments (estimator model) or 2) the distance between the better hypothesis and the sources/references (translation ranking model). However, they tend to be more resource hungry and therefore less relevant for low-resource scenarios.

4.1.2 Low-resource machine translation

Low-resource machine translation is the same task applied to situations called 'low-resource', where only little training data is available. This ranges from languages with a limited number of native speakers, such as extinct (e.g. Medieval French), endangered (e.g. Aromanian) or vulnerable (e.g. Alsatian) languages, to situations for which little data is available, such as 'unusual language pairs' (e.g. translating from German to Italian is less common than from French to English, for example), or 'unusual data domains' (e.g. medical datasets are rarer than newspaper datasets).

The architectures used are the same, but several papers comparing SMT with NMT in very low-resource settings conclude that SMT performs better, being more accurate and less prone to overfitting (Bollmann 2019; Dowling et al. 2018; Singh and Hujon 2020; Skadiņa and Pinnis 2017). Sennrich and Zhang (2019) analysed and reproduced previous comparisons, to conclude that SMT can actually be outperformed by NMT when architectures and hyper-parameters are carefully chosen, but only above a certain quantity of data. Performance, however, is still considerably lower than in high-resource situations.

4.2 How do we increase performance?

Several techniques allow to mitigate low-resourced situations. In this section, we will introduce a subset of the easiest to implement and most frequent techniques used, which will serve as a basis for our own data augmentation experiments. For more exhaustive surveys of existing techniques, we refer the reader to Haddow et al. (2021), Stahlberg (2020), and Wang et al. (2021).

4.2.1 At the data level

In MT when studying a low-resourced language pair (such as French-Alsatian), it is possible to leverage information from a higher resourced language pair (such as French-German), or to use more available data of a different nature (parallel lexicons) or of a different type (monolingual data). Some research even investigates the use of multimodal data (data combining several modalities, such as image and text, text and video, and so forth).[3]

[3] Multimodality usually implies changing the chosen model architectures, and it is out of scope for this document.

Extending parallel data Existing datasets can be augmented using domain information to transform existing sentences into new data, for example by replacing words by similar terms (grammatically or semantically) (Hasigaowa and Wang 2019), using paraphrase systems (Fadaee et al. 2017; Marton et al. 2009; Xu et al. 2020), or transliteration from a third language (Libovický et al. 2020). This can add some noise to the data; however, noise, when in measured quantities, has been proven to increase the robustness of models during training, and augmenting datasets (not only parallel) with noise is therefore an available strategy (Edunov et al. 2018; Lample et al. 2017; Li et al. 2020a; Li and Specia 2019b). Parallel datasets have also been extended with monolingual data 'as such' for training (Currey et al. 2017).

Using monolingual data For low-resourced languages, it is easier to find monolingual datasets than parallel datasets, and this monolingual data can be leveraged to create new parallel sets using a technique called **backtranslation** (Sennrich et al. 2016a).[4] In our Alsatian-French example, Alsatian is the low-resourced language: after having trained, even badly, a MT model on the rare Alsatian to French data available, we can use it to translate our monolingual low-resourced data (Alsatian) to the other language (French). This creates a new dataset of noisy French (artificial) to quality Alsatian, which can then be added to train a model on French to Alsatian. This operation can then be repeated iteratively, to hopefully generate parallel sets of increasing quality, which is then called **iterative backtranslation** (Hoang et al. 2018). Applying these methods in the opposite direction is called forward translation (Abdulmumin et al. 2020). Many variations on this theme exist, from combining back and forward-translation (Libovický et al. 2020), combining back-translation with pivoting[5] (Xia et al. 2019), focusing on back-translating harder target words (Fadaee and Monz 2018), using these methods for domain adaptation (Bertoldi and Federico 2009), and so forth (Bojar and Tamchyna 2011). Edunov et al. (2018) provide a survey of their advantages and limitations. All the newly created datasets can then be used, mixed with "real" parallel data, for **training** (Marie et al. (2020) suggest adding a tag to differentiate back-translated data from real data when training) or for **pretraining**, to initialize models weights before training on the task of interest. Pretraining can be done using **monolingual pretraining** sets, using source to source/target to target sentences to constrain the language modeling capabilities of the encoder/decoder (Khatri and Bhattacharyya 2019) or artificial parallel data such as **back-translated pretraining** sets (Currey et al. 2017).

Using auxiliary language data A special case of pretraining, called **pivot-learning**, uses data from a high-resourced language pair to initialise the full model, and hope information from the high-resourced pair can be used by the low-resourced pair (Casas et al. 2018; Casas et al. 2019; Li et al. 2020b; Luo et al. 2019; Sánchez-Cartagena et al. 2019; Xia et al. 2019). In our example, it could be using French to German or English to initialize French to Alsatian. If the model is not trained after and used as such (for example, if there is no training data for the low-resourced pair), this is called **zero-shot** transfer (Aharoni et al. 2019; Lauscher et al. 2020; Vergés Boncompte and R. Costa-jussà 2020). Some experiments have also been done to compare using average-ressourced languages related to the low-resource language of interest, versus unrelated high-resource languages as pivots, and Kocmi and Bojar (2018) use high-resourced

[4] It is also possible to use monolingual datasets of source and target languages to induce lexicons and learn a common latent space between them in an unsupervised way (Artetxe et al. 2017; Hu et al. 2019; Khatri and Bhattacharyya 2019; Lample et al. 2017), but we will not consider unsupervised methods in this document as they usually need a new model architecture.

[5] Pivot-learning is detailed in next paragraph.

unrelated languages as pivot with very good results. Libovický et al. (2020) compare oversampling monolingual data, backwards translation, forward translation, using a pivot-language to generate synthetic data, and transfer-learning from the pivot.

Multilinguality in models Simulatenously multilingual models are a special case of using language data, and they have been shown to help low-resource scenarii by providing data in other languages and constraining the hidden representations to a shared, language-independent space. The amount of sharing between languages varies depending on the approach, from architecture changes using multi-encoder, multi-decoder architectures (Luong and Manning 2016), optionally sharing attention mechanisms (Firat et al. 2016b), to single model approaches with a single shared encoder and decoder(Johnson et al. 2017).[6]

Using data from other fields Data from other tasks (e.g. morphosyntactic) can also be integrated in models to constrain translations (which often implies changing the model architecture), for example dependency trees, used as soft-templates to constrain translations (Yang et al. 2020b), POS tags, predicted jointly with translations (Burlot et al. 2017), and so forth. These avenues can be more costly in terms of data for very low-resourced fields, and we will not investigate them.

4.2.2 At the model level

We only mention these methods in passing, as, though they represent an important development of the field, they also usually rely either on the development of 'non-standard' architectures or the integration of resource-intensive models.

With language models Model-based data augmentation methods include using pre-trained multilingual models, such as XLM (Lample and Conneau 2019) or mBART (Liu et al. 2020) to improve machine translation (Liu et al. 2020; Siddhant et al. 2020). Pre-trained language models can also be integrated directly in the translation model (Gulcehre et al. 2015), and translation models can also be pre-trained with a language modeling or denoising objective (Guo et al. 2020).

Embeddings management A non negligible part of efforts have been focused on embedding management, to mitigate the out-of-vocabulary word problem, particularly prevalent in low-resource machine translation. Methods range from combining character and word level embeddings (Ataman et al. 2019; Chen et al. 2018; Luong and Manning 2016) or creating hierarchical embeddings (Morishita et al. 2018; Yang et al. 2020a) and combining models with dictionnaries or similarity scores to look up or find most similar words to unknowns (Luong et al.

[6] Multilingual models can seem close to pivot-learning models, but the former learn to encode and decode several languages all at the same time (using training data which clearly indicates the current example language pair), where the latter have been trained on given languages, then are fine-tuned on new languages (in a sequential process), without any indication to the model that the language pairs have changed between both steps.

2015b; Zhang et al. 2013; Zhang et al. 2012), to using multilingual embedding spaces to manage out of vocabulary words (Alvarez-Melis and Jaakkola 2018; Duong et al. 2016; Gouws and Søgaard 2015; Haddad et al. 2018; Li et al. 2016; Zou et al. 2013).

Examples of other architecture changes Several efforts have been made to find more complex architectures, ranging from simply plugging encoder embeddings directly into decoder attention mechanisms (Ngo et al. 2019) to constraining the hidden representation to make it a latent variable (Zhang et al. 2016), or even learning language graphs to determine optimal path between two languages (He et al. 2019); these are out of scope for this document, as we focus on simple and well-known methods, in an optic of usability and understandability both by machine learning practicioners but also linguists.

4.3 Conclusion

In this last theoretical chapter, we saw what differentiates machine translation in high and low resource setups, as well as many methods to mitigate scarce data. Among these, we can remember data augmentation through pretraining, backtranslation, and multilinguality, as well as transfer learning, which we will use in some of our experiments. It is now time to jump into the heart of our work, and look at our first axis: using neural networks to study words through time. Onwards!

Part II

NEURAL NETWORKS AND COGNATES

> The angles were vaguely illogical. It was like a puzzle. This red bead, if slid along this wire to that junction, should reach there—but it didn't. A maze, odd, but no doubt instructive.
>
> Padgett (1943)

Character-level low-resource machine translation and cognate prediction present a number of similarities: both model sequence-to-sequence relations, looking at structured data, and machine translation has been used with for cognate prediction or proto-form reconstruction in recent years. Preliminary experiments (Fourrier 2020) and previous work (Dekker 2018) have shown that sequence-to-sequence models do not perform as well as simpler neural or non-neural methods on etymological cognates. This raises the question which started this PhD:

Can we efficiently use neural networks for historical word prediction at all?

Hypothesis The neural networks tested might have failed on this task for several plausible reasons. 1) Cognate prediction could be theoretically too different from machine translation, as their underlying linguistic assumptions and aims have not been formally compared, and could impact the transferability of choices and techniques. 2) The models used in preliminary work could be inefficient because of a lack of adaptation to the task: this is likely, as most have been used 'out-of-the-box', with no parameter search and no fine-tuning. 3) Neural networks are known to be inefficient in very low-resource situations, being easily outperformed by statistical methods, and cognate prediction is notoriously low-resource.[1] 4) Cognate prediction could simply be too complex a task for these methods.

Setup To differentiate between these hypotheses (Chapter 6), we first study the theoretical differences between cognate prediction and machine translation, to suggest likely good setups for this task. Then, to test the 'model' hypothesis without being hindered by a lack of data, we create artificial datasets which reproduce phonological language change, and study if neural networks can reach an acceptable accuracy for artificial historical word prediction, and under which conditions. We then loop back to the linguistic aspect of the task, by wondering what experiments on artificial data can teach us about historical word prediction.

Having established model performance in an ideal setup, as well as the minimal data size theoretically needed, we then wonder how transferable our results are to real datasets. After updating an etymological database to generate cognate sets big enough to train on (Fourrier and Sagot 2020b), our next question is the following:

Which low resource machine translation setups work best on real historical word prediction?

[1] For example, the most famous cognate sets (manually annotated by linguists) are the historical Swadesh (1955) lists, which contain at most about 200 word pairs per language.

Hypothesis The machine translation field has developed a number of tools to mitigate low-resourced situations, among which pretraining, backtranslation, and exploiting model multilinguality are highly used (see Section 4.2.1). But whereas machine translation models could theoretically be trained on any sentence pair that are translations of each other, cognate prediction is far more limited in terms of which data can be used; cognacy relations only link a limited number of words in specific language pairs, limiting not only available parallel data but also the potential for synthetic data (e.g. via backtranslation or pretraining using monolingual lexicons). However, since cognates are words linked by sound correspondences and sound change rules identified across multilingual sets, we expect multilinguality to be a plus.

Setup We therefore reproduce the experiments done on artificial data, using this time real data from Romance languages, then try a panel of data augmentation techniques, and compare their respective performance on our task (Chapter 7). Having then determined which setups are the best, we study what the corresponding models are able to learn linguistically, as raw performance on such a task is of little value if it does not rely on linguistic generalizations. Are the models only learning to repeat the most common patterns in the data, or do we see more complex linguistic observations emerge?

Our final question is the following:

WHAT can we learn by probing cognate prediction models?

Hypothesis Cognates descend from a common ancestor word, their proto-form. When models learn mappings between cognates in related languages (going from all languages to themselves through all possible pairs), the multilingual joint intermediate representation is constrained to a common denominator, since each encoder needs to be coherent for all decoders, and each decoder to be able to work on representations coming from any encoder. A plausible candidate for this common space would be a mapping of a shared ancestor space, as proto-form have the overall smallest distance to all their children.

Setup We therefore train a massively multilingual cognate prediction model, this time without explicit ancestor information, and study both what it learns externally (by analysing results as such), and internally (by probing and representing its inner components) (Chapter 8).

5 Experimental setup

> The sun is simple. A sword is simple. A storm is simple. Behind everything simple is a huge tail of complicated.
>
> Pratchett (2010)

Having developped in the previous introduction our different research hypotheses and broad research directions, we describe in this chapter the practical setup of our experiments: which models we use, how they are implemented,[1] which scores we choose for evaluation and why, as well as the datasets we train on (as almost all datasets used in this manuscript were developped especially for this research, from artificial data to test our hypothesis to automatically extracted cognate sets). We will refer back to this chapter in all chapters of the current manuscript part.

5.1 Models and training

Task description Our general task here will be historical word prediction, either including both cognate prediction and proto-form reconstruction, or only the former. We model this task as a character-level[2] machine translation task, on cognate sets.[3]

Baseline Our baselines are SMT models trained for each bilingual language direction (**SMT**), using `MOSES` (Koehn et al. 2007),[4] with `GIZA++` as tokenizer and alignment model (Och and Ney 2003), a 3-gram `KenLM` model of the output (Heafield 2011), and `MERT` for tuning (Bertoldi et al. 2009).[5]

Neural models We use our implementation of the sequence-to-sequence encoder-decoder model with attention (Bahdanau et al. 2015; Luong et al. 2015a) in `fairseq`,[6] and `fairseq`'s implementation of the multilingual Transformer (Vaswani et al. 2017).

[1] However, detailed parameters for given model are described in their respective chapters.
[2] We use here the customary term "character-level MT," although in our case, characters correspond to phones.
[3] It is actually more similar to a word-level MT, where we consider our sentence to be our cognate, and our words to be its phones.
[4] MOSES is the reference library for statistical MT.
[5] See section 4.1.1 for more explanation of statistical machine translaton models.
[6] Code can be found at `http://github.com/clefourrier/CopperMT`.

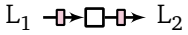
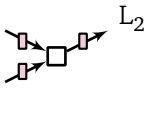
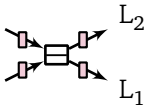
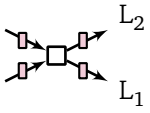
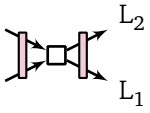
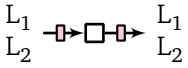
Name	#source	#target	With monolingual data	Sharing components	Schematic
SMT	1	1	No	-	
B-NMT	1	1	No	None	
B-NMT+m	1	1	Yes	None	
M-NMT	9 (all)	9 (all)	No	None	
M-NMT+m	9 (all)	9 (all)	Yes	None	
M-NMT+m+shared_emb	9 (all)	9 (all)	Yes	Embeddings	
M-NMT+m+shared_all	9 (all)	9 (all)	Yes	All	

Table 5.1: Model type setups - NMT can be a RNN (NMT^R) or a Transformer (NMT^T).

Several setups will appear throughout this document: bilingual NMT models, without (**B-NMT**) or with (**B-NMT+m**) added monolingual data,[7] and multilingual models without (**M-NMT**) or with (**M-NMT+m**) monolingual data, using one encoder and one decoder per language. We also use multilingual models with shared components, either embeddings layers[8] (**M-NMT+m+shared_emb**) or full encoders and decoders across all languages (**M-NMT+m+shared_all**).[9] Recurrent models are signified using NMT^R, and Transformers NMT^T.

Each model is trained using the Adam optimizer (Kingma and Ba 2015) and the cross entropy loss, stopping on the first of either 20 epochs or convergence, using dev BLEU as criterion during training.

Evaluation We use BLEU as an evaluation metric on characters,[10] using the SacreBLEU implementation (Post 2018). In order to use BLEU even when we produce n>1 "translations", we compute BLEU scores by providing the n-best results as the references, and our input word as the output, which provides an estimation of the "best performance" across "translations".[11] In standard MT, BLEU can under-score the many valid translations that do not match the refer-

[7] B-NMT+m models train on a single language pair, augmented with the monolingual target data, provided to the decoder through its own encoder; they allow the target decoder to see as much target data as possible, to reinforce its language modelling capacities.

[8] Sharing the embedding layer means sharing the embedding vector space and vocabulary across all languages encoders, same across decoders.

[9] When sharing the model across languages, we prepend our data with a target language token.

[10] Several other character-based metrics could also have been used, but we focused on the standard metric for machine translation.

[11] This informs us more about recall rather than precision.

ence. For cognate prediction, however, we expect a single correct prediction in most cases (there are a few exceptions such as variants due to gender distinctions specific to the target language). This makes BLEU better suited to the cognate prediction task than it is to standard MT.[12]

Note on our neural models Our first experiments were done using our first implementation of the recurrent encoder-decoder models with attention named MEDeA (Multiway Encoder Decoder Architecture) and based on PyTorch directly. However, for the sake of reproducibility and improved usability, our implementation was since then ported to `fairseq`, one of the most used toolkits in NLP. For better homogeneity throughout the thesis, the preliminary experiments have been rerun with the new implementation (Chapter 6); new results are more exhaustive, and statistically comparable to the initial results, which can be found in the following papers (Fourrier 2020; Fourrier and Sagot 2020a).

5.2 Data

In this section, we introduce two dataset types: artificial data built to simulate natural cognate evolution from protoform to several daughters in Romance languages in a controlled setup, and real data, either historical or contemporary, extracted from real Romance languages. Both are contributions of this PhD: the artificial data generator created for the occasion is open-source software, and provided with the set of rules extracted from the litterature for Romance languages, and the different cognate sets come from an etymological database updated for this thesis, also provided as open source software with extraction scripts.

5.2.1 Generating artificial datasets

Using artificial data for such a proof of concept offers several advantages: we can investigate the minimum number of word pairs required to successfully learn sound correspondences (though in a very simplified and noiseless setup), as well as control the different parameters constraining the proto-language (number of phonemes, phonotactics) and its transformation into the daughter languages (e.g. number of sound changes). However, the artificial data must be realistic, to not impair the linguistic validity of the experiment; the proto-language must have its own realistic phonology, obey phonetic and phonotactic rules, and its daughter languages must have been generated by the sequential application of plausible sound changes.

Creating a proto-language We create an algorithm which, given a phone inventory and phonotactic constraints[13], generates a lexicon of a chosen size.[14][15] For our experiments, we draw inspiration from Latin and Romance languages. More precisely, we use:

[12] BLEU is also more adapted than an exact match, as it allows us to compare how close the prediction is to the reference, and does not suffer, in cognate prediction, from the same problems as in standard MT.

[13] Phonotactics govern which phonemes sequences are allowed.

[14] We do not consider plausible morphological variations of the generated forms.

[15] Code available at `http://github.com/clefourrier/PLexGen`.

- The phone inventories of Romance languages: each lexicon generated uses all the phones common to all Romance languages, as well as a randomly chosen subset of less common Romance phones.[16]
- The phonotactics of Latin, as detailed in the work of Cser (2016): each word is constructed by choosing a syllable length in the distribution, and its syllables are then constructed by applying a random set of the corresponding positional phonotactic rules.

Generating daughter languages Given the proto-language, we create a daughter language by, first, randomly choosing a set of sound changes, then consecutively applying each chosen sound change to all words in the lexicon. Among the main possible sound changes for Romance languages are epenthesis (addition of a sound to a word, as in Latin *tremulare* giving French *trembler*), of which prothesis is a subset (addition of a sound at the beginning of a word, often a vowel for Romance, such as Latin *status* to Spanish *estado*), its opposite, apocope (the loss of a final vowel, as in Latin *mare* giving Portuguese *mar*), as well as palatalisation (moving a consonant or vowel closer to the palate during its pronunciation, such as Latin *clamare* to Spanish *llamar*), lenition (changes in consonant manner which increase the air flow), and diphtongisation (going from one to several vowels). The dataset generated for this paper used two sets, each of 15[17] randomly chosen sound changes, to generate two daughter languages. Three examples from our generated dataset are: 1) [stra] to [isdre], [estre]; 2) [ʒolpast] to [ʒolbes], [ʒolpes] and 3) [splutoi] to [isbledoi], [espletoi].

5.2.2 Extracting real historical data

Languages Sound correspondences and sound change rules are identified by looking at multilingual sets of cognates. We select 9 related Romance languages for which enough cognate data is available, and their common parent language: Galician (GL), Portuguese (PT), Spanish (ES), Catalan (CA), Occitan (OC), Italian (IT), French (FR), Romanian (RO), Aromanian (RUP) and finally Latin (LA).[18]

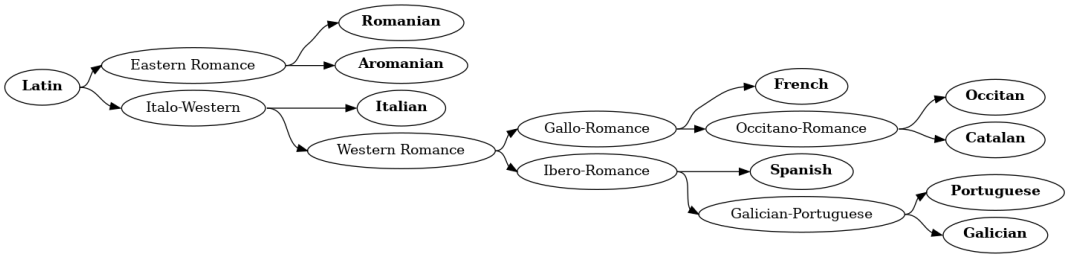


Figure 5.1: Relations between studied languages and their families.

The Romance family divided early in two branches (Figure 5.1): the Eastern Romance branch (RO, RUP), and the Italo-Western branch (all others). They therefore constitute the two oldest language clusters in our data. However, through external influences on their phonology,

[16] For example, vowels common to all Romance languages are [a] [e] [i] [o] [u], and a subset of extra vowels could be [ɔ] [ɛ] [ɪ].

[17] Preliminary experiments showed that 15 sound changes allowed the daughter languages to diverge enough, while still exhibiting the similarities we expect in our synthetic data.

[18] We also considered adding Sardinian (SC) and Dalmaian (DLM) to our studied languages, but had to resort not to, due to a lack of data and plausible phonetizers.

French (Germanic influences) and the Eastern Romance branch (Slavic influences) tend to diverge from the other Romance languages studied. At the opposite end of the spectrum in terms of language closeness, Portuguese and Galician belong to their own language sub-branch, the Galician-Portuguese branch, as do Catalan and Occitan in the Occitano-Romance branch.

Source and extraction EtymDB2 is a database of lexemes, stored as triples of the form ⟨language, lemma,[19] meaning expressed by English glosses⟩, which are related by typed etymological relations (such as 'direct inheritance' 'borrowing' and 'cognacy', derivtional relations being ignored[20] see Section 1.2.2.2). To generate the cognate dataset from EtymDB2, we followed the inheritance etymological paths between words. Two words form a cognate pair if they share a common ancestor in one of their common parent languages.

EtymDB2 is an update of the database EtymDB1, both extracted from the Wiktionary, which has been done during this work in order to gather enough cognate data. This update work and subsequent analysis on phylogeny have been summarised in an LREC paper (Fourrier and Sagot 2020b), and will not be developped here.

Cognacy relations, in general, link lexemes; however, this extraction process selects the available lemmas as being the representative form of their lexemes for our cognate pairs. These lemmas might not necessarily be the best form to observe cognacy and sound correspondence patterns, as they can have lost some important phonetic information on the way: for example, ***blanc*** [blɑ̃] 'white (masc.)' (French), coming from ****blancus*** [blankos] 'id.' (Vulgar Latin, reconstructed) no longer carries the sound correspondence between Latin [k] and French [ʃ] that is still observable in the feminine form ***blanche*** [blɑ̃ʃ] 'white (fem.)' (French). Similary, French ***écrire*** [eʁʁiʁ] 'to write' coming from Latin ***scrībere*** [skri:bere] 'id.' does not highlight as many correpondances as other forms, such as ***écrivons*** [eʁʁive] '(we) write', where the [b] to [v] correspondence is clearly visible.

Throughout our extraction, lemma are only kept based on availability: for Latin, the lemma of a noun lexeme is often its nominative singular form, but we sometimes also find accusative singular forms; for adjectives, it is often the nominal masculine form, but we also find nominal neutral, and nominal feminine forms, sometimes accusative or dative. In the case of a cognate pair where one word is associated with several counterparts (say a Spanish lexeme associated with the nominative and accusative forms of its corresponding Latin cognate), the pair kept is the one with the lowest phonetic Levensthein distance (see next paragraph).[21]

Preprocessing We first **clean** the dataset, and remove words containing characters that are not in the correct locale for the relevant language. We **filter** the remaining list: when faced with competing pairs, i.e. pairs whose source word is the same but whose target words differ, we retain the pair with the lowest Levenshtein edit distance. Each word is **phonetised** into IPA using `espeak` (Duddington 2007-2015), an open source multilingual speech synthesiser which can also phonetize sequence of words to IPA, in CA, ES, IT, FR, LA, PT, RO. We approximate the

[19] See Section 1.1.2 for a definition.

[20] Lexemes linked to their parent by derivation processes do not always carry the full phonetic evolution of their languages: derivation can add affixes at a later stage of the word evolution from its parent form, affixes which do not necessarily carry this full phonological history (depending on their creation mechanism). In order to better focus on words carrying the full path of sound changes, we ignored derivational lexemes during the database creation.

[21] Other possible setups could have been either 1) keeping all pairs carrying these parallel relations, and making sure every cognate pair carried all possible forms, to better help the model learn ambiguity, but it would have implied creating a new metric for evaluation of several predictions against several baselines, as well as a huge amount of time, or 2) choosing an homogeneous, default lemma for a POS and a language (singular feminine form for French nouns for example), with the risk that some finer or rarer sound correspondences would not be seen by our models.

Language	Ancestor path
Aromanian	Proto-Indo European >Proto-Italic >Old Latin >Latin
Catalan	Proto-Indo European >Proto-Italic >Old Latin >Latin >Old Occitan >Old Catalan
French	Proto-Indo European >Proto-Italic >Old Latin >Latin >Old French >Middle French
Galician	Proto-Indo European >Proto-Italic >Old Latin >Latin >Old Portuguese
Italian	Proto-Indo European >Proto-Italic >Old Latin >Latin
Latin	Proto-Indo European >Proto-Italic >Old Latin
Occitan	Proto-Indo European >Proto-Italic >Old Latin >Latin >Old Occitan
Portuguese	Proto-Indo European >Proto-Italic >Old Latin >Latin >Old Portuguese
Romanian	Proto-Indo European >Proto-Italic >Old Latin >Latin
Spanish	Proto-Indo European >Proto-Italic >Old Latin >Latin >Old Spanish

Table 5.2: Ancestors of our languages of interest in our chosen database.
Latin as an ancestor corresponds both to Classical Latin (written) and Vulgar Latin (spoken). See Table A.3 in Appendix for relevant languages wiktionary codes.

phonetization of OC as CA, RUP as RO, and GL as PT.[22] This also means we consider only one 'standard' pronunciation for each of our languages (dialectal versions would highlight similar if slightly different sound correspondence patterns). We then homogeneize to remove accentuation marks and homogeneise double consonant representations. The data is then **tokenised** at the character level with a manually implemented rule-based phonetic tokenizer, which keeps symbols (diacritics for nasal vowels, symbols of length) with the corresponding phone.

For example, *conocer* 'to know' is phonetised as [konoθɛɾ], then split into phones and segmented into [k, o, n, o, θ, ɛ, ɾ], and *cattus* 'cat' is phonetised as [kat:us], which gives, after tokenisation [k, a, t:, u, s].

Did we obtain cognate datasets? Despite data sanity checks (by comparing the source etymological database with previous works), cleaning steps, and the focus on 'inheritance' relations and language ancestry paths, it is still very likely that the generated datasets contained misclassified words. Strictly asserting that our datasets only contained cognates and absolutely no borrowings would have required a time and expertise I did not have.

5.2.3 Extracting monolingual lexicons

Non-historical words Monolingual lexicon datasets are used for data augmentation experiments in our historical word reconstruction task only. They are extracted from a multilingual translation graph, YaMTG (Hanoka and Sagot 2014), by keeping all unique words for our languages of interest (IT, ES, LA). To remove noise, words containing non-alphabetic characters are discarded (punctuation marks, parentheses, etc.).

Historical words Where the previous dataset used monolingual lexicons containing any type of word from an historical linguistic standpoint, this time, we generate monolingual cognate lexicons for our 'massive cognate prediction experiment'. They contain only words that

[22] These approximations should hold for our study, as these languages have the most linguistic features in common.

could belong to cognate sets, as they descend from a direct parent of their language (for example, Latin or Old Spanish for Spanish) - these words are therefore likely to contain phonological patterns of interest.

5.2.4 Data description

We now introduce the three datasets created for each task of interest. To first study feasability of using neural networks for our task, we use controlled artificial data; then, to study the different tasks of historical word reconstruction (cognate prediction, proto-form reconstruction), we use a small specific real data set, with added monolingual lexicon for data augmentation experiments; lastly, to investigate what our models learn during cognate prediction specifically, we use a massively multilingual dataset.

Artificial dataset The full set contains 20,000 unique word triples with a proto-language (PL) word and its reflects[23] in the two daughter languages (DL1 and DL2). Samples of various sizes are then randomly drawn from this dataset.

Historical word reconstruction To study the applicability of our task to real dataset, we use bilingual cognate lexicons linking Spanish Italian and Latin,[24] extended with monolingual lexicons for some experiments (see Table 5.3). The cognate and proto-form lexicons respectively contain 5,109, 4,271 and 1,804 words for a total of 77,771, 63,131 and 24,576 phones (ES–IT being considerably smaller), with on average 40 different and unique phones. The final monolingual lexicons (cleaned and phonetised, extracted from a dictionary) contain between 18,639 and 99,949 unique words (the LA set is more than 4 times smaller than the others).

Bilingual	LA–IT	LA–ES	ES–IT
#words	5,109	4,271	1,804
#phones	77,771	63,131	24,576
#Unique phones	34	39	38
Avg. word length	7.62	7.40	6.81
Monolingual	ES	IT	LA
#words	78,412	99,949	18,639
#phones	626,175	815,562	142,955
#Unique phones	38	40	29
Word length	7.98	8.24	7.67

Table 5.3: Dataset statistics for our lexicons.

Massive cognate prediction To study what a neural network latently learns from using only contemporary data, we develop a massive Romance cognate set. It contains considerable variability in the number of word pairs between languages (see Table 5.4): OC→RUP (two of our least resourced languages) contains 81 pairs, whereas PO→ES contains 1,930 pairs. Monolingual

[23] In other terms, its 'children' words.

[24] We use an extended definition of cognacy which includes the proto-form.

datasets vary from 553 words for OC to 6,005 words for IT, CA, ES, FR, IT, and PT monolingual sets contain more than 2000 words, and GL, OC, RO and RUP less than 1,500.[25] The total number of phones per pair varies accordingly; the number of unique phones per language pair stands between 32 and 56, depending on the number of shared phones between languages. Average word length varies between 5.3 and 8.3 phones.

5.2.5 Datasets use

These datasets will be successively used in the following chapters

- first, the artificial dataset will help us determine whether neural networks can learn historical word reconstruction in a noiseless and easy setup in the next chapter;
- then, the historical word reconstruction dataset will help us test if neural networks can learn historical word reconstruction in a real setup in the following one;
- finally, the massive cognate dataset will help us see what and how precisely our models can learn from the task of cognate prediction specifically.

[25] We use monolingual data to reinforce the decoders language modelling capabilities, see next section. We expect that such a variation in size will impact learning.

From Catalan (CA) to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
#words	2,612	1,233	466	449	970	324	1,031	235	144
#phones	16,472	16,171	5,706	5,724	12,511	3,486	13,601	2,162	1,307
#unique phones	36	41	47	44	56	35	44	42	40
Avg word length	7.31	7.56	7.12	7.37	7.45	6.38	7.60	5.60	5.54
From Spanish (ES) to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
#words	1,236	4,967	693	732	1,880	230	1,930	463	291
#phones	16,198	34,176	8,931	9,760	25,686	2,534	26,156	4,700	2,898
#unique phones	41	35	46	44	54	38	44	42	39
Avg word length	7.55	7.88	7.45	7.67	7.83	6.51	7.78	6.08	5.98
From French (FR) to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
#words	466	694	3,772	215	715	110	600	135	86
#phones	5,707	8,941	21,225	2,641	9,332	1,126	7,665	1,183	737
#unique phones	47	46	46	42	54	41	43	37	36
Avg word length	7.13	7.44	6.63	7.15	7.53	6.12	7.39	5.39	5.30
From Galician (GL) to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
#words	449	732	215	1,464	558	138	882	176	106
#phones	5,724	9,759	2,641	9,509	7,196	1,455	11,117	1,703	1,005
#unique phones	44	44	42	35	51	41	37	38	37
Avg word length	7.37	7.67	7.15	7.50	7.45	6.27	7.30	5.84	5.74
From Italian (IT) to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
#words	973	1,885	717	558	6,005	234	1,557	618	378
#phones	12,534	25,742	9,346	7,190	44,073	2,660	21,199	6,834	4,046
#unique phones	56	54	54	51	49	50	55	50	47
Avg word length	7.44	7.83	7.52	7.44	8.34	6.68	7.81	6.53	6.35
From Occitan (OC) to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
#words	324	230	109	138	234	553	222	117	81
#phones	3,486	2,534	1,120	1,455	2,659	3,026	2,391	1,044	724
#unique phones	35	38	41	41	50	33	42	38	36
Avg word length	6.38	6.51	6.14	6.27	6.68	6.47	6.39	5.46	5.47
From Portuguese (PT) to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
#words	1,031	1,930	596	883	1,556	223	4,891	399	261
#phones	13,606	26,158	7,624	11,125	21,188	2,399	33,046	3,991	2,569
#unique phones	44	44	43	37	55	42	37	39	38
Avg word length	7.60	7.78	7.40	7.30	7.81	6.38	7.76	6.00	5.92
From Romanian (RO) to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
#words	236	465	136	175	621	117	398	1,088	412
#phones	2,173	4,715	1,193	1,696	6,859	1,044	3,984	5,833	4,251
#unique phones	42	42	37	38	50	38	39	32	32
Avg word length	5.60	6.07	5.39	5.85	6.52	5.46	6.01	6.36	6.16
From Aromanian (RUP) to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
#words	146	292	87	107	378	81	259	412	817
#phones	1,327	2,907	745	1,015	4,038	724	2,551	4,251	4,531
#unique phones	40	39	37	37	47	36	38	32	29
Avg word length	5.54	5.98	5.29	5.74	6.34	5.47	5.92	6.16	6.55

Table 5.4: Detailed dataset statistics for our lexicons.

6 Can machine translation neural networks be used for historical word prediction?

> "Messengers are on their way to your household at this moment, Khadilh ban-harihn," he said. "We regret the delay, but it takes time, you know. All these things take time."
>
> Elgin (1969)

Now that we know what our experimental setup will look like, we want to know whether the previously seen machine translation tools can actually be used for historical word prediction. Preliminary work (Dekker 2018) seemed to conclude that these models were too big or complex for such a task, and we therefore first study cognate prediction from a theoretical standpoint, in order to better understand the extent of its similarities and differences to machine translation. Having reached a conclusion on what this could theoretically mean for our current question, we want to test if machine translation techniques can be used to predict cognates, using artificial data and exploring different model parameters. This allows us to control both the amount of data available as well as the model level of complexity, and therefore to pinpoint if models previously failed on the task because they were too complex, the data too small, or if it is actually a matter of theoretical mismatch between task and chosen architecture.

6.1 Theoretical comparison between cognate prediction and machine translation

We first study if the underlying linguistic assumptions and aims of machine translation and historical word prediction are distinct, and how it could impact technique transfer from the first task to the latter.

6.1.1 Form

Units The first obvious difference is that machine translation processes sentences split into individual **graphemic** units, where cognate prediction, on the other hand, involves predicting sound correspondences from one cognate word to another, and so is best modelled using se-

quences of **phones**. This is only a surface difference, in terms of neural networks models – apart from the fact that, where machine translation could rely on pretrained vocabulary embeddings, no such thing exists for phonetic embeddings. Therefore, vocabulary size should be similar to that of a character-level translation model.

Order In machine translation, correspondences between source and target sentences can involve long-distance reorderings, in character sequences of sentence length, whereas the reorderings sometimes found in the correspondence between cognates are almost always local (e.g. metatheses), and involve character sequences of word length.[1] This introduces a fundamental difference, from a machine learning point of view, between both tasks, and it is unlikely that the best performing architectures for machine translation, able to model these long-distance relations across sentences, will also be the best for cognate prediction – and it seems respectively logical that short-distance bi-directional architectures could outperform more complex models (as sound changes can both occur after or before specific sounds, as in assimilation cases).

Data size As mentionned earlier, cognate sets are considerably smaller than usual machine translation sets, by several orders of magnitude. It is very likely that this will play on model size, as a model with typical parameters (several layers in encoders and decoders, of hidden size 512, which are default in several toolkits, such as `fairseq` and PyTorch)[2] would be too big for so small a task, and would risk severly overfitting.

6.1.2 Substance

Modeled relations Machine translation involves symmetrical relations between sequences, as does cognate prediction when looking at sister languages. However, when adding proto-form reconstruction to the mix, we introduce asymmetrical relationships: parent-to-child, i.e. modelling sequences of regular sound changes, is non-ambiguous, whereas child-to-parent (e.g. ES→LA) is intrinsically ambiguous, as two distinct sounds in the parent language can result in the same outcome in the child language. When two distinct sounds in the child language are the outcome of the same sound in the parent language, it is always because their (word-internal) phonetic contexts were different in the parent language. In other words, the parent-to-child direction is (virtually) non-ambiguous, but might require taking the phonetic context into account. However, the child-to-parent direction is intrinsically ambiguous, which results from the fact that a sound in the child language can be the regular outcome of more than one sound in the parent language: for instance Spanish /b/ comes from Latin /p/ in *abría* (from Latin *aperīre*) but from Latin /b/ in *habría* (from Latin *habeō*). This introduces the need to find methods to accommodate this ambiguity, unnecessary in 'pure' machine translation, in order to predict both correct and plausible answers. We suggest that using n-best predictions can increase performance significantly.

There is no apparent reason that machine translation models should not be able to model

[1] Even with different segmentation granularities for MT, the average sequence length is generally much shorter for cognate prediction than for MT.

[2] See the documentation of the `fairseq` LSTMs at `https://fairseq.readthedocs.io/en/latest/models.html` or of the PyTorch Transformer at `https://pytorch.org/docs/stable/generated/torch.nn.Transformer.html`.

historical word prediction. Higher performing models are likely to be considerably smaller than usual MT models (apart from vocabulary size), good at modelling short-term bidirectional relations, and to predict several n-best results. Let's therefore try to understand why previous models did not work.

6.2 Can MT be used for historical word reconstruction?

This chapter is an extended and more exhaustive version of (Fourrier 2020) and (Fourrier and Sagot 2020a).

6.2.1 Experimental setup specificities

In order to study precisely the conditions under which our translation models can be used for cognate prediction, we artificially create an 'ideal setup', with controlled data and data size. We use our artificial lexicon, composed of a proto-language and its reflect in two artificially defined daughter languages.

Data Since we want to study the impact of data on our performance, we extract several subsets from our artificial set (of 500, 1000, 1500, 2000, and 3000 words, see Section 5.2.1), that we shuffle using 3 different data seeds, and split in training/development/testing sets (respectively using 85/7.5/7.5% of the data). We therefore run our experiments three times, one on each seed, to be statistically significant.

Models Our baseline is the SMT model. We use B-NMT and M-NMT recurrent (NMT^R) and Transformer (NMT^T) setups (see Section 5.1).

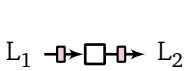
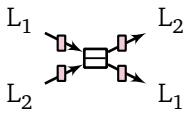
B-NMT	M-NMT
	

Table 6.1: Models used.

Since we want to find the best setup for each situation, we perform a hyperparameter search for all our models (separately). We run optimisation experiments for all possible bilingual and multilingual architectures, using three different data splits for each parameter combination studied, and choosing the models performing best across seeds. Our initial parameters were selected from preliminary experiments (in bold in Table 7.2).

	Parameters	Values studied
1)	Learning rate × Batch size	{0.01, **0.05**, 0.001} × {10, 30, **65**, 100}
2)	Embed. dim. × Hidden dim.	{8, 12, 16, **20**, 24} × {18, 36, **54**, 72}
3)	Number of layers	**1**, 2, 4
Transformers - 4)	Number of heads	**1**, 2, 3, 4
RNNs - 4)	Attention type	None, Bahdanau, Luong (**dot**, concat, general)

Table 6.2: Parameter exploration experiments for NMT models.
In bold, the initial parameters at each step.

Table 7.2 contains the successive parameter exploration steps: at the end of a step, we automatically selected (according to average dev BLEU) the step-best value (after discarding un-

stable combinations, identified by their dev BLEU standard deviation above 6), used as input parameter for the next parameter exploration step. When looking at multilingual models, we chose the model performing best on most languages, as measured by comparing the sum of the ranks (according to their average performance per language) of each model over all language pairs.

6.2.2 Raw best results

We first describe the best performing models results.

Language pair	SMT	B-NMT^R	B-NMT^T	M-NMT^R	M-NMT^T
			500 word pairs		
PL→ D1	98.6±1.7	96.6±2.1	87.1±1.4	97.5±2.3	79.4±1.6
PL→ D2	98.7±1.8	98.4±2.3	70.8±5.5	99.0±1.0	75.7±0.7
D1→ D2	86.7±6.1	91.5±3.0	80.2±1.7	93.7±0.9	76.5±3.8
D2→ D1	97.6±0.7	95.2±2.7	88.8±0.4	96.1±2.5	81.5±1.4
D1→ PL	65.7±2.6	60.0±2.7	50.2±4.1	65.1±2.4	50.5±3.3
D2→ PL	67.1±1.7	68.6±2.6	44.5±4.0	70.0±3.0	54.3±7.8
			1000 word pairs		
PL→ D1	97.9±0.6	98.6±1.3	89.9±2.5	98.1±1.4	92.8±2.4
PL→ D2	97.7±1.2	98.5±0.7	87.0±1.1	97.8±1.7	94.0±1.7
D1→ D2	85.8±2.6	93.1±2.0	88.5±3.6	93.8±2.5	89.3±1.4
D2→ D1	95.7±0.8	95.8±1.0	92.8±1.8	95.6±1.6	92.0±1.3
D1→ PL	62.7±4.6	62.7±5.5	53.2±4.8	59.4±3.0	60.0±2.2
D2→ PL	65.0±3.6	66.5±1.8	54.7±3.0	63.1±2.9	61.1±2.7
			1500 word pairs		
PL→ D1	98.9±0.0	99.6±0.4	93.7±0.5	99.0±0.9	94.1±2.1
PL→ D2	98.5±0.4	99.7±0.2	90.5±1.4	99.2±0.8	92.8±2.0
D1→ D2	88.1±1.8	92.3±0.7	89.4±0.5	94.6±1.3	90.4±1.3
D2→ D1	98.2±0.6	98.2±0.5	95.3±1.6	98.3±0.2	93.8±2.2
D1→ PL	62.6±0.5	65.7±0.8	62.3±2.3	65.9±0.4	61.4±1.1
D2→ PL	65.6±1.5	67.1±3.3	64.9±2.3	68.1±1.1	64.2±1.8
			2000 word pairs		
PL→ D1	98.1±0.3	98.7±0.4	94.4±0.4	98.7±0.7	96.2±0.6
PL→ D2	98.0±0.5	98.9±0.8	93.6±0.6	98.8±1.0	94.0±1.3
D1→ D2	88.5±1.8	95.0±0.7	91.8±0.8	93.6±1.5	93.1±0.7
D2→ D1	97.5±0.8	97.4±0.9	96.4±0.7	96.9±0.4	95.8±0.4
D1→ PL	61.7±1.6	65.2±0.8	62.9±1.1	65.0±1.7	64.0±2.9
D2→ PL	63.6±0.5	64.7±1.7	63.5±0.9	67.4±0.8	65.3±1.8
			3000 word pairs		
PL→ D1	99.0±0.1	99.2±0.2	97.0±0.7	99.4±0.5	96.6±1.1
PL→ D2	98.4±0.4	98.9±0.4	94.5±0.1	99.3±0.4	96.0±1.0
D1→ D2	88.0±0.4	92.9±0.4	90.5±1.3	92.6±1.0	91.1±1.9
D2→ D1	97.0±0.9	97.5±0.9	96.9±0.6	97.2±0.5	96.2±1.3
D1→ PL	61.7±0.7	63.7±1.7	63.0±2.3	64.8±1.0	61.4±2.1
D2→ PL	65.9±0.2	66.7±1.1	64.0±0.4	67.4±0.9	65.3±2.9

Table 6.3: Test BLEU raw results for our best parameters.

SMT We observe, on Table 6.3, that the SMT best models reach 97.7 to 99 BLEU when predicting children from the protoform (PL→D1 and D2), 85.8 to 98.2 when predicting one children to the other (D1⟷D2), and 61.7 to 67.1 when reconstructing the protoform from the childrens (D1 or D2→PL). We can therefore say that it is possible, on simple enough data, to predict cognates and reconstruct protoforms using statistical machine translation. We then observe that increasing data size does not change accuracy in a statistically significant way for SMT, only increasing the prediction stability (reducing the standard deviation between runs, to less than one for our biggest dataset).

The reasons for variations between translation directions will be looked at in Section 6.4.

B-NMT^R Our B-NMT^R models reach 96.8 to 99.7 BLEU for the prediction of children from the protoform, 92.1 to 98.1 when predicting cognates, and 57.9 to 68.7 for protoform reconstruction: an optimized recurrent NMT model can therefore also both predict cognates and reconstruct protoforms. As for SMT, increasing data size only seems to increase stability.

B-NMT^T Our B-NMT^T models, however, reach 87.6 to 97.0 BLEU for the prediction of children from the protoform, 88.0 to 96.8 when predicting cognates, and 45.5 to 64.5 for protoform reconstruction. The average performance of Transformer NMT models is lower than for the other bilingual models, and these models seem more susceptible to the effect of data size, gaining 10 BLEU points from the smallest to the largest setup.

M-NMT^R Our M-NMT^R models directly reach a similar or better performance than SMT for the lowest data size (between -1/+1 BLEU to +7 BLEU), as well as perform better than B-NMT^R (between +1 to +7 BLEU for 500 data pairs). Increasing data size to 1500 allows to reach the best results of the table across data pairs. Multilingual recurrent models therefore seem to be the best possible architecture for cognate prediction and protoform reconstruction – however, these models are slower to train.

M-NMT^T Again, Transformer models, this time multilingual, underperform for the lowest data sizes, by 10 to 15 BLEU points compared to SMT, reaching a similar performance to B-NMT^T; their performance remains correlated for bigger data sizes, underperforming the other models.

6.2.3 Conclusion

As we just saw, all machine translation models tested can, when chosen properly, predict cognates and reconstruct protoforms linked by simple enough relations. SMT and NMT^R models outperform NMT^T models, which correlates with state-of-the-art observations on the fact that Transformers models underperform in low-resource situations when compared to recurrent models. It is interesting to note that SMT performs as well, using default parameters, as NMT^R models optimized for the task. We also observed an asymetry in prediction depending on the task, which we will study in Section 6.4. But first, we want to know how to best choose our hyperparameters for these tasks.

6.3 How can MT best be used for historical word reconstruction?

We now study the results of our hyperparameters search, to determine which parameters are the most relevant for our task. We first look at the different parameters of interest for our bilingual setups, then multilingual setups, before summarizing the trends of interest. All results summarised here are presented as results table in Appendix B, with each cell representing an experimental setup applied to 3 dataset splits.

The tables in Appendix B represent 1638 experiments.

6.3.1 Bilingual experiments results

6.3.1.1 B-NMT^R

Best parameters Absolute best parameters vary in terms of embedding dimension, hidden size, and batch size, but almost always use a learning rate of 0.005, and Luong's attention, 1 or 2 layers (see Table 6.4).

Parameter trends Appendix B.1 contains the detailed results for our experiments. The main first trend we observe across all tables is the impact of the data size: using 500 word pairs to learn on is consistently too small to allow stable learning for most random parameters combinations. Across language pairs, average BLEU scores reach from 13.2 to 95.1 when varying embedding dimensions and hidden sizes, and 5.3 to 96.7 when varying batch size and learning rate. However, carefully tuning embedding dimension to hidden layer and learning rate to batch size ratios allows the bilingual models to reach, even with such a small set, a similar performance to models trained with more data (above 60 BLEU for the daughters to proto-language directions, above 90 for the other language directions). More generally, for our artificial datasets, we reach peak performance above 1500 data pairs, where almost all parameter combinations tried perform in the same range and without too much variation (standard deviations on average below 1.5).

Then, we can observe that the different parameters need to be explored differently. First, best results are obtained when hidden size and embedding dimensions are jointly increased, staying within a ratio where hidden dimension is three times the size of the embedding dimension (with the exception of the embedding size 12 to hidden dimension 90 which performs surprisingly well) (Appendix B.1.1).[3] Then, too small a learning rate (0.001) prevents learning, especially with small datasets; however, using too big a learning rate without decreasing the batch size does not work well either: best results are obtained with 0.005 as learning rate, and batch sizes between 30 and 60 (Appendix B.1.2). Using a small number of layers works well, but 4 layers prevent learning, with BLEU scores on average 20 points lower for 4 layers and less than 1500 data pairs (Table B.7 in Appendix); this is most likely because it increases model size too much with respect to the data. In a similar vein, the best performing attention is Luong's dot attention (the smallest in terms of parameters), quickly matched by the other attentions from Luong's paper with enough data (more than 1500) (Table B.9 in Appendix).

[3] This is easily observable with the apparition of a diagonal across the experiment matrices.

Model	Embedding dim.	Hidden dim.	Batch size	Learning rate	#layers	Attention
			500 word pairs			
PL→ D1	28	90	30	0.005	2	Luong dot
PL→ D2	24	72	30	0.005	2	Luong dot
D1→ D2	28	90	30	0.001	2	Luong dot
D2→ D1	28	90	100	0.005	1	Luong dot
D1→ PL	28	90	65	0.01	1	Luong dot
D2→ PL	28	90	30	0.001	1	Luong general
			1000 word pairs			
PL→ D1	12	90	65	0.005	1	Luong dot
PL→ D2	12	54	65	0.005	1	Luong dot
D1→ D2	16	36	65	0.005	1	Luong dot
D2→ D1	16	54	30	0.005	1	Luong general
D1→ PL	12	90	30	0.005	1	Luong dot
D2→ PL	24	72	130	0.005	1	Luong dot
			1500 word pairs			
PL→ D1	16	72	30	0.005	1	Luong dot
PL→ D2	16	54	65	0.005	1	Luong general
D1→ D2	28	54	130	0.01	2	Luong general
D2→ D1	28	72	65	0.005	2	Luong dot
D1→ PL	20	72	65	0.005	1	Luong general
D2→ PL	28	72	100	0.005	1	Luong dot
			2000 word pairs			
PL→ D1	20	54	30	0.005	2	Luong dot
PL→ D2	20	36	65	0.005	1	Luong dot
D1→ D2	20	90	65	0.005	1	Luong dot
D2→ D1	20	90	100	0.005	1	Luong concat
D1→ PL	28	54	100	0.01	1	Luong dot
D2→ PL	12	54	65	0.005	2	Luong concat
			3000 word pairs			
PL→ D1	16	90	30	0.005	1	Luong dot
PL→ D2	16	90	100	0.005	2	Luong concat
D1→ D2	24	72	30	0.005	1	Luong dot
D2→ D1	12	54	130	0.01	2	Luong concat
D1→ PL	28	54	100	0.01	1	Luong dot
D2→ PL	16	90	130	0.005	1	Luong general

Table 6.4: Results of parameter exploration experiments for B-NMT^R.

6.3.1.2 B-NMT^T

Best parameters Absolute best parameters vary in terms of hidden size, but almost always use the maximum embedding dimension of 28, a learning rate of 0.005, 1 head, and 1 or 2 layers, with a lower batch size for smaller datasets (see Table 6.5).

Parameter trends At maximum data size (3000 word pairs), Transformers still don't reach BLEU stability across hyper-parameters, varying for example between 76 and 94.7 for

Language pair	Embedding dim.	Hidden dim.	Batch size	Learning rate	#layers	#heads
500 word pairs						
PL→ D1	28	36	30	0.005	1	1
PL→ D2	28	72	65	0.005	1	1
D1→ D2	28	72	30	0.01	1	1
D2→ D1	28	90	30	0.005	1	1
D1→ PL	28	36	65	0.005	1	1
D2→ PL	24	54	65	0.01	1	3
1000 word pairs						
PL→ D1	28	72	30	0.005	1	1
PL→ D2	28	72	30	0.005	2	1
D1→ D2	28	72	30	0.005	1	2
D2→ D1	28	72	30	0.005	1	1
D1→ PL	28	90	65	0.005	1	1
D2→ PL	28	54	65	0.005	1	1
1500 word pairs						
PL→ D1	24	90	30	0.005	2	1
PL→ D2	28	36	30	0.005	2	1
D1→ D2	24	90	30	0.005	1	1
D2→ D1	28	54	30	0.005	1	1
D1→ PL	28	72	30	0.005	2	1
D2→ PL	28	36	30	0.005	2	1
2000 word pairs						
PL→ D1	28	36	30	0.005	1	4
PL→ D2	28	72	30	0.005	2	4
D1→ D2	28	72	100	0.005	1	1
D2→ D1	28	72	65	0.005	2	1
D1→ PL	28	72	65	0.005	2	1
D2→ PL	28	72	65	0.005	1	1
3000 word pairs						
PL→ D1	28	72	65	0.005	2	4
PL→ D2	28	90	30	0.005	1	2
D1→ D2	28	36	65	0.005	1	2
D2→ D1	28	36	65	0.005	2	1
D1→ PL	28	36	100	0.005	2	2
D2→ PL	28	90	65	0.005	1	2

Table 6.5: Results of parameter exploration experiments for B-NMT^T.

PL→D1's comparison of hidden to embedding size. Below this threshold, BLEU scores vary even more, for example between 1.1 and 73.5 for the same word pair (Appendix B.1.1). Transformers are considerably more impacted than recurrent models by the lack of data.

However, some results can still be observed. When studying embedding size versus hidden dimension, the best results are consistently obtained by increasing embedding size only, with maximum BLEU (at 60 for daughters to proto-language, 90 to 95 for the rest for the maximum amount of data) obtained for an embedding size of 28 (Appendix B.1.1).[4] When studying batch

[4]This can be observed with the apparition of a pattern of increasing vertical stripes across our matrixes.

size to learning rate (Appendix B.1.2), it is better to use a bigger learning rate with a smaller batch size for the smaller datasets (with best results obtained at 0.001 to 30 for 500 word pairs), then decrease the learning rate to 0.005 with enough data (above 1500 word pairs, included). Using one layer gives the best results for lower datasizes, two give equivalent results above 1500 data pairs, but four is, as for the RNNs, consistently too big, most likely by increasing model size too much (Table B.7 in Appendix). Lastly, varying the number of heads has no impact on the results (Table B.8 in Appendix).

6.3.2 Multilingual experiments

Best parameters The best results this time are most of the time 1) for the M-NMT^R models, an embedding dimension of 24 for a hidden dimension of 72 to 90, various batch sizes with a learning rate of 0.005, one or two layers and the Luong dot attention and 2) for the M-NMT^T models, the maximum embedding dimension of 28 against varying hidden dimensions, 30 or 65 batch size against mostly 0.005 learning rate, 2 layers and 1 head (see Table 6.6).

Model	Embedding dim.	Hidden dim.	Batch size	Learning rate	#layers	Model specific
		M-NMT^R				Attention
500	24	72	30	0.005	1	Luong dot
1000	24	90	30	0.005	1	Luong dot
1500	16	72	65	0.005	2	Luong dot
2000	24	90	130	0.005	2	Luong dot
3000	24	90	100	0.005	2	Luong concat
		M-NMT^T				#heads
500	28	36	65	0.01	1	1
1000	28	72	30	0.005	2	1
1500	28	54	30	0.005	2	1
2000	28	90	30	0.005	2	1
3000	28	90	65	0.005	2	2

Table 6.6: Results of parameter exploration experiments for M-NMT^R.

Parameter trends Overall, parameter tendencies are comparable to those of bilingual models (Appendix B.2). For M-NMT^R, better results are obtained by increasing embedding dimension and hidden size jointly, when for M-NMT^T, we only need to increase embedding dimensions. For both, the learning rate must not be too small to reach convergence, and for smaller sets, best results are obtained with lower batch sizes. Less layers still provides best results, though this time, 4 layers gives comparatively good results above 1500 word pairs, likely because multilingual models see enough data to train these extra layers (1500 word pairs times 6 translation directions). For M-NMT^R, Luong's dot is still the best attention, though this time, with enough data (3000 word pairs), it is also possible to reach good performance without any attention; for M-NMT^T, the number of heads does not change results, except for the lowest data sizes, when too big a head number gives bad results.

6.3.3 Synthesis

Important hyperparameters For our task, the most important parameters to study, both for our NMT^T and NMT^R models are batch size and learning rate. Then, NMT^T models need a high embedding dimension, and NMT^R models an embedding size around 1/3 of the hidden size (above a minimal size), with one of Luong's attention (dot if little data).

Impact of model types We observed that NMT^R and SMT models were systematically better than NMT^T models. They also were comparatively more stable on average (lower BLEU standard deviations), and from a lower data size than Transformers. NMT^R and SMT models seem more adapted to our task.

Impact of multilinguality Parameter trends are similar when looking at bilingual and multilingual models, though final best parameters differed. If this is applicable to real data too, it could save time by looking at preliminary trends on bilingual models (smaller and faster to train), to make educated guesses about best multilingual parameter trends, and study a restrained parameter subset. Multilinguality did not seem to have such an impact on best results, but it reached a better and more stable performance sooner for 'bad' parameter combinations.

Impact of data size From 1000 to 1500 data pairs, NMT^R models are overall quite stable, and manage to reach good performance for most hyper-parameter combinations chosen – NMT^T, however, are more susceptible to data size, and are still not stable at 3000 word pairs. SMT models perform as well at 500 data pairs as they do at 3000. For the smallest data sizes we will reach, it is plausible that SMT will perform better than neural models, but NMT^R models should not be too far behind.

6.4 What do our experiments teach us about historical word reconstruction?

As we saw in Section 6.2, not all prediction directions behave the same. Going from the protolanguage to its daughter reaches a test BLEU between 95 and 99 in the best setups, predicting from one daughter to another reaches a lower test BLEU, between 90 and 98, while going from the daughters to their parent has the lowest test BLEU, between 60 and 70. This reflects the intuition we developped with respect to the specificities of modelled relations in cognate prediction, developed in Section 6.1.2: going from parent language to daughter, being unambiguous, is easier than the other way around, which is filled with ambiguity. To accommodate this ambiguity, we study the use of predicting n-best results, and not just the best one, in an attempt to allow the model to predict other plausible answers.

Using the best models for each language pair and model type, we predict the 1 to 10-best test results to compute related BLEU. We display our results on Figure 6.1, where the grid columns indicate the model type (NMT^R in orange, NMT^T in blue, and SMT in green), and grid rows indicate the language pairs: the first two lines display prediction from protolanguage to daughters

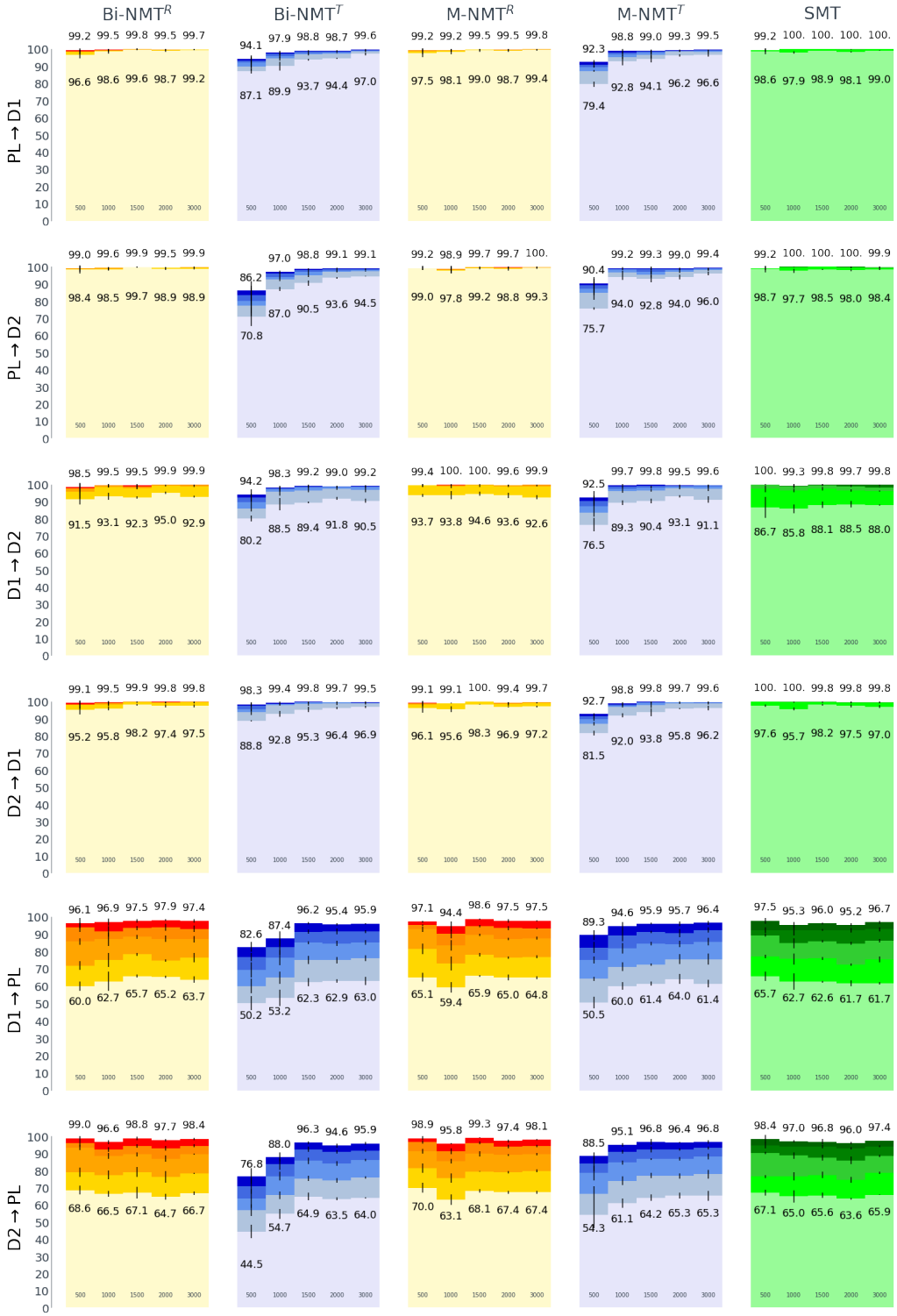


Figure 6.1: Test BLEU scores for our experiments on artificial data.
Colours indicate the model type: NMT^R in orange (bilingual in col 1 and multilingual in col 3), NMT^T in blue (bilingual in col 2 and multilingual in col 4), SMT in green (col 5).
Colour shades indicate the value of n in n-best predictions (1, 2, 3, 5 and 10 from bottom to top).
The numbers (x-axis) indicate the data size, from 500 to 3000.

(the least likely setup when working with real data, except in low-resourced situations where data from an ancestor is more common than data from the language of interest), the next two are daugther to daughter, and correspond to the usual cognate prediction task, then the two last are daughters to protoform, corresponding to the task of protoform reconstruction. The grid's cell themselves contain 5 columns, one for each data size studied, with a shading gradient indicating the n value of the n-best prediction (1, 2, 3, 5, 10 from pastel to bright shades, bottom to top).

We first confirm our previous observations on test BLEU ranges depending on the concerned languages data size available (500 triplets performing worse than 1000+), as well as the fact that adding data only benefits the Transformer. We then observe that n-best seems to solve the problem precedently evoqued: going from 1 to 3-best allows to earn on average 30 extra BLEU points in our most ambiguous situation, protoform reconstruction (last two grid rows). Going to 5 or 10-best allows to reach similar performance than for non-ambiguous situations, but at the cost of precision. When using automatic historical word prediction as a tool to aid linguists, as in Bodt et al. (2018), the aim is not to predict *the* single correct answer, but to provide a list of plausible candidates, which can then be tested against field reality or monolingual historical corpora.[5] As a side note, we also observe that, for the protoform reconstruction, multilingual models seem to outperform bilingual models, particularly in 1 and 2-best, especially in low-resourced situations (500 data pairs). Mutlilinguality might be an interesting way to mitigate scarce data situations for protoform reconstruction, and we hypothesize that this could be linked to the way our M-NMT models learn and constrain their intermediate representation, which will need to be studied later.

We therefore conclude that, when working with real data, it will likely be important to take into account the difference in ambiguity, due to the variety in prediction direction and tasks (with the most ambiguous task being protoform reconstruction and the least descendant prediction, with cognate prediction in between), and to balance this ambiguity using n-best predictions or multilinguality.

6.5 Conclusion

In this chapter, we learned that all models, when properly tuned, can learn the correspondances we are interested in in a controlled setup - though quite a simplified one, with extremely regular and noiseless data. However, recurrent and statistical machine translation models outperform Transformers, especially for small data sizes. We also painted a picture of the most important parameters to optimize. Lastly, we introduced n-best prediction as a tool to manage ambiguity in prediction directions when studying historical word reconstruction, and confirmed that it helps, as well as that some prediction directions will always be harder than others because of the inherent ambiguity they contain (protoform reconstruction). Our main interrogation now is how this all transfers to actual historical data, which contains added noise linked to languages 'real world' evolution, and this is the focus in next chapter.

[5] This aim is quite different from the situation in machine translation, where we often only need one correct translation (hopefully the best ranked by the model), and where n-best predictions provide also correct variations of the best translation. In historical word prediction, however, **at most** one prediction is correct while other predictions, compatible with the phonetic laws involved, could have been correct but are not. A linguist would be interested in both correct and plausible predictions, not just the best ranked one.

7 Do low resource machine translation setups work on real historical word prediction?

> "The only thing to do with it is feed it to the computer." She fed it to the computer which ate it with evident pleasure.
>
> Lafferty (1974)

Having established that historical word prediction can, in an ideal setup, be modelled as a machine translation task, we decide to extend this work to real-world historical data. Real data is likely to be noisier than the artifical data we previously used, as well as smaller, and we want to see how this will impact learning. We first study our previous machine translation architectures in this real setup, using Romance languages for our hyperparameters search. We then study if cognate prediction can actually benefit from low-resource translation techniques, and compare the impact of different data augmentation methods common in low resource machine translation (backtranslation, pretraining, multilinguality).

This chapter is an extended version of Fourrier et al. (2021). Both our code and data are freely available at `github.com/clefourrier/CopperMT`. Results differ slightly - within significance margins - because a number of the experiments were re-run.

7.1 Experimental setup

7.1.1 General setup

Data We use the 'historical word reconstruction' dataset, containing Latin (LA) and its children, Italian (IT) and Spanish (ES). We run all experiments on three different train/dev/test splits in order to obtain confidence scores. For the bilingual (baseline) and multilingual setups, each split is obtained by sampling sentences 80%/10%/10% randomly.

Models Our baseline is the SMT model. We use B-NMT and M-NMT recurrent (NMT^R) and Transformer (NMT^T) setups (see Section 5.1).

B-NMT	M-NMT
$L_1 \rightarrow L_2$	$L_1 \rightarrow L_2$, $L_2 \rightarrow L_1$

Table 7.1: Models used.

Since we want to find the best setup for each, we perform hyperparameter searches.

7.1.2 Preliminary hyperparameters search

We seek to determine whether MT architectures and techniques are well suited to tackling the task of cognate prediction, paying attention to avoid the pitfalls raised by Sennrich and Zhang (2019) by carefully selecting architecture sizes and other hyper-parameters. We run optimisation experiments for all possible bilingual and multilingual architectures, using three different data splits for each parameter combination studied, and choosing the models performing best across seeds. Our initial parameters were selected from preliminary experiments (in bold in Table 7.2), just like in the previous chapter.

	Parameters	Values studied
1)	Learning rate × Batch size	{0.01, **0.05**, 0.001} × {10, 30, **65**, 100}
2)	Embed. dim. × Hidden dim.	{8, 12, 16, **20**, 24} × {18, 36, **54**, 72}
3)	Number of layers	**1**, 2, 4
Transformers - 4)	Number of heads	**1**, 2, 3, 4
RNNs - 4)	Attention type	None, Bahdanau, Luong (**dot**, concat, general)

Table 7.2: Parameter exploration experiments for NMT models.
In bold, the initial parameters at each step.

Table 7.2 contains the successive parameter exploration steps: at the end of a step, we automatically select (according to average dev BLEU) the step-best value, used as input parameter for the next parameter exploration step.[1]

7.1.3 Main task setup

For our baselines, we use our SMT models with the best performing B-NMT^R and B-NMT^T models. We then assess the impact of techniques commonly used to improve MT in low-resource scenarios. We first investigate the impact of using monolingual data for all 3 architecture types, via pretraining and backtranslation,[2] then compare these data augmentation methods with our best performing multilingual NMT models. We finally combine our best performing methods.

Monolingual pretraining For NMT, one way to take advantage of additional monolingual data is to teach the model to "map" each language to itself by using an identity function objective (i.e. learning to copy the input word: the model will learn to predict as output the input it is shown) on the monolingual data for the model's target language. For the multilingual model, this means that every encoder will see data from all languages, whereas each decoder will only see data from its specific language. Using monolingual target data during pretraining will mostly allow each target decoder to have seen more target data (which avoids overfitting and helps specializing them on their languages). We expect it to be beneficial to encoders too, since

[1] When looking at multilingual models, we chose the model performing best on most languages, as measured by comparing the sum of the ranks (according to their average performance per language) of each model over all language pairs.

[2] For another explanation of these methods, see Section 4.2.1

our source and target languages tend to share common sound patterns in cognate prediction, being closely related. In practice, we pretrain the model for 5 epochs[3] using the identity function objective together with the initial cognate prediction objective (on the original bilingual data) and then fine-tuned on the cognate task as before for 20 epochs. For SMT, model parameters cannot be pretrained as in NMT, so in the guise of pretraining, we take the nearest equivalent: we use target-side monolingual data to train an extra language model. For each language pair, the monolingual dataset we use is composed of 90% of the target monolingual data. The bilingual data is the same as before.

Backtranslation For each architecture type, we use the previously chosen SMT and B-NMT models to predict 10-best results for each seed from the monolingual target-side data, and construct synthetic cognate pairs from monolingual lexicons and source-side predictions. For each word, we keep the first prediction of the 10 that also appears in the relevant monolingual source language lexicon as our new source, and the initial source as target (this is akin to filtering back-translated data (e.g. to in-domain data) in MT, a standard practice). We discard pairs with no prediction match. This large back-translated bilingual dataset is extended with our original training set (see 4.2.1 for a layman explanation of back-translation). For NMT, it is used to train a new model for 10 epochs,[3] which is then fine-tuned for 20 epochs with the original bilingual training set. For SMT, it is used (instead of the original bilingual data) to train a new phrase table (which is then used jointly with the bilingual phrase table learned for the pair).

7.2 Hyperparameter search results

7.2.1 Bilingual models

All parameters are summarized in Figure 7.1, with full colored tables with numbers in Appendix C.1.

7.2.1.1 Impactful parameters

Embedding dimension vs hidden size Similar to our experiments on artificial data, for the RNN, increasing embedding dimension has little impact on its own, but increasing hidden layer dimension does. For Transformer models, it is the opposite, as increasing embedding dimension is more important than varying hidden layer dimension (see Figures 7.1, column 1).

Batch size vs learning rate We observe a similar behaviour between all models: learning rate and batch size must vary cohesively, with higher batch sizes correlated with higher learning rates. A good average is 0.005 to 65 for the Transformer, and 0.005 to 30 for the encoder decoder. The Transformer is less sensitive to imbalanced ratios with too small learning rates to too big batch sizes, and the RNN to too big learning rates to too small batch sizes (Figure 7.1, column 2).

[3] This number of epochs is systematically big enough to reach convergence.

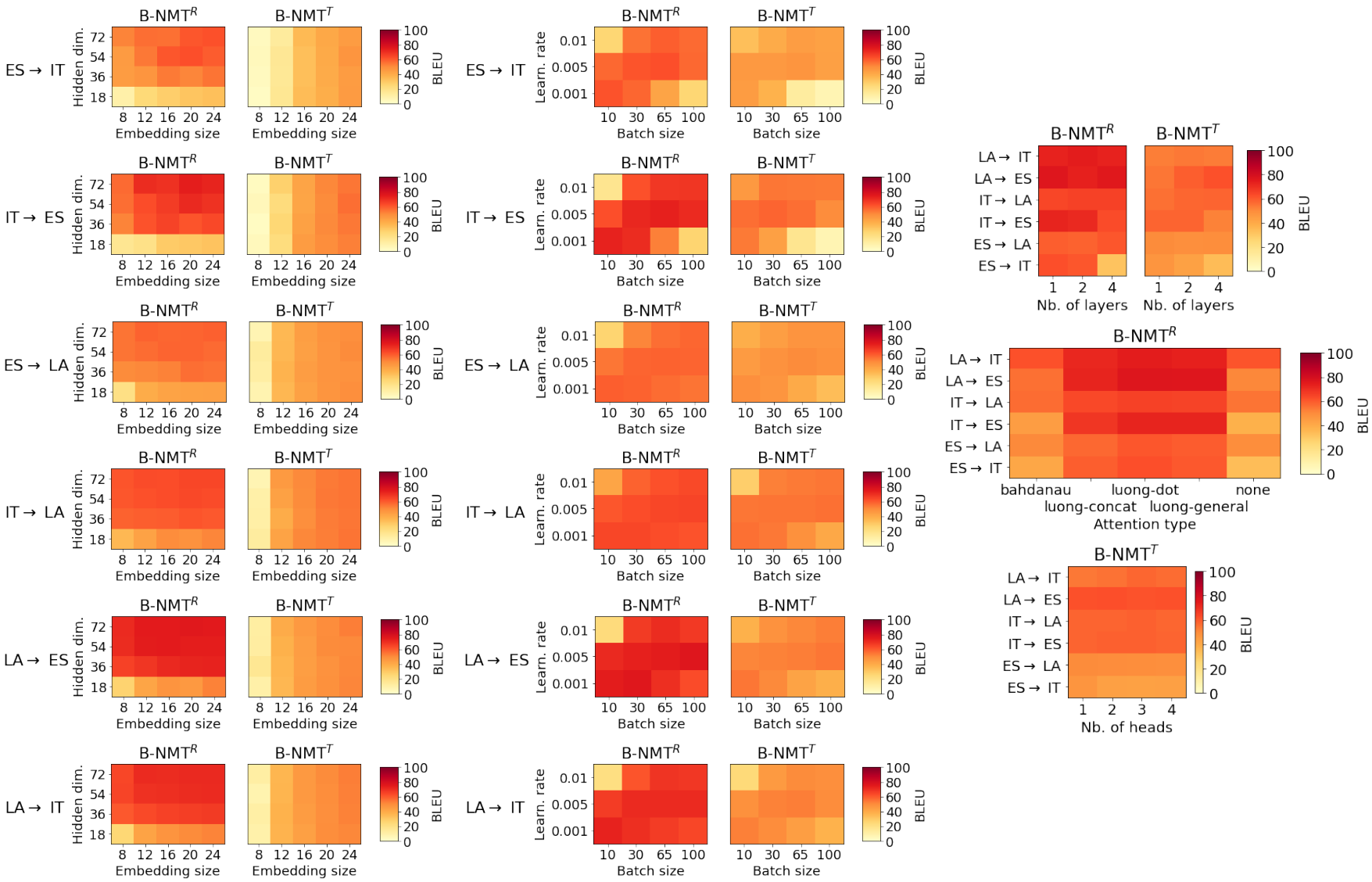


Figure 7.1: Synthesized results of B-NMT hyperparameter search (Development BLEU of the best checkpoint).

RNN attention type All possible attentions from the works of Bahdanau et al. (2015) and Luong et al. (2015a) were used; attentions from Luong et al. (2015a) reach better results, with the Luong dot attention slightly over performing the others (Figure 7.1, column 3, 2nd image).

7.2.1.2 Less significant parameters

Number of layers For both models, the number of layers does not change the performance, as long as the number of epochs is increased to reach convergence (Figure 7.1, column 3, 1st image).

Transformer heads number The head number does not have a statistically significant impact on the best Transformers' accuracy (Figure 7.1, column 3, 3rd image). However, for models with worse parameter combinations, using 4 heads increases instability and decreases convergence rate.

7.2.1.3 Overall best parameters

The final parameters chosen are detailed in Table 7.3, and reflect similar trends as with our artificial data, confirming the validity of our previous conclusions.

Model	Learning rate	Batch size	Embed. dim	Hidden dim	#layers	Model specific
B-NMT^R						Attention type
ES→IT	0.005	65	20	54	1	Luong-dot
IT→ES	0.005	65	20	72	1	Luong-dot
ES→LA	0.001	10	24	72	4	Luong-dot
LA→ES	0.005	100	20	72	1	Luong-dot
IT→LA	0.001	10	24	72	2	Luong-dot
LA→IT	0.001	10	20	72	2	Luong-dot
B-NMT^T						#heads
ES→IT	0.005	65	24	54	1	1
IT→ES	0.005	30	24	54	1	3
ES→LA	0.005	65	24	54	1	2
LA→ES	0.001	10	24	72	4	2
IT→LA	0.001	10	24	72	4	3
LA→IT	0.005	65	24	72	2	3

Table 7.3: Results of parameter exploration experiments for RNN and Transformer models.

7.2.2 Multilingual models

Hyperparameter trends are the same between bilingual and multilingual models. All parameters are summarized in Figure 7.2, with full colored tables with numbers in Appendix C.2.

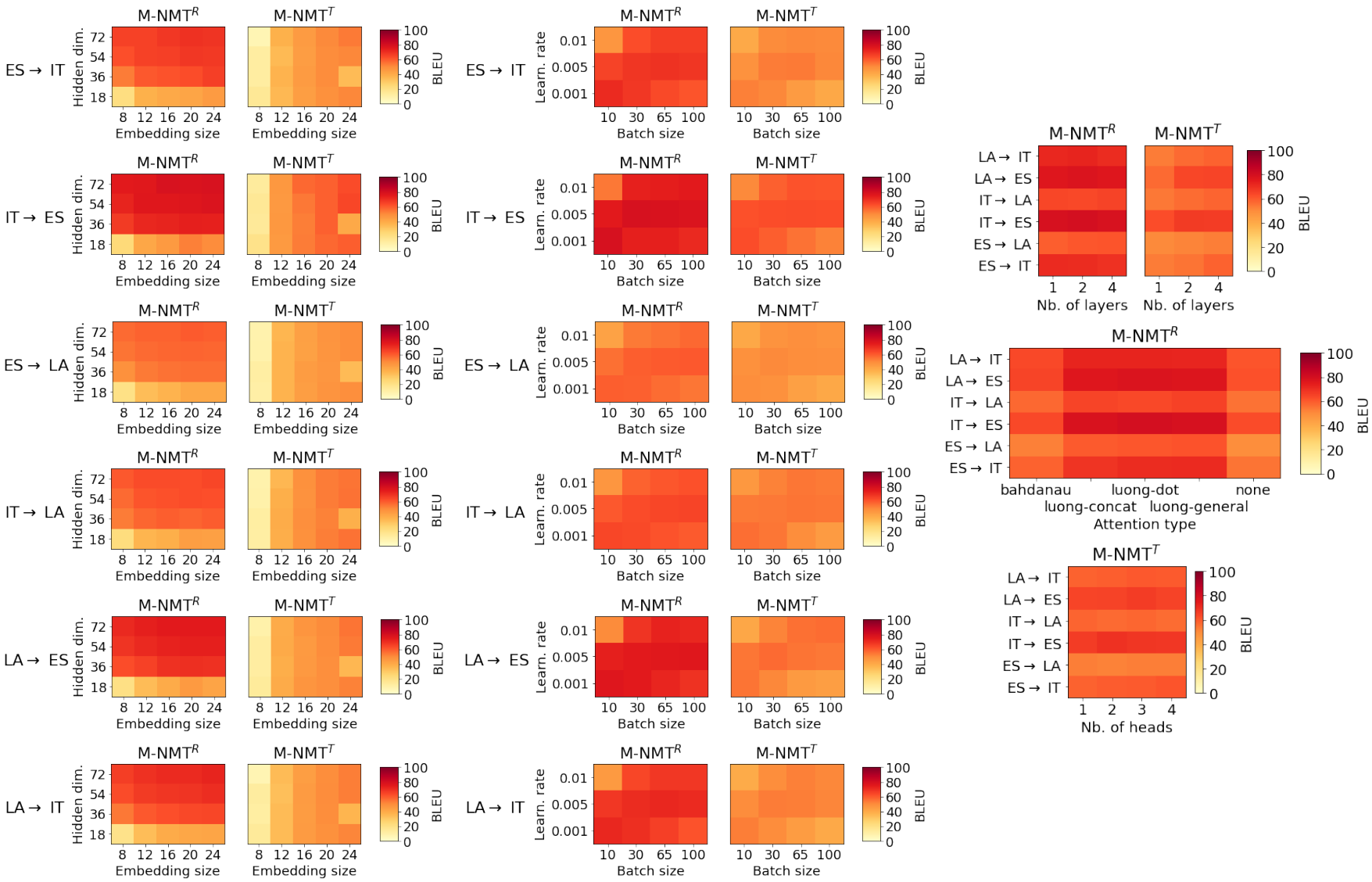


Figure 7.2: Synthesized results of M-NMT hyperparameter search (Development BLEU of the best checkpoint)

7.2.2.1 Impactful parameters

Embedding dimension vs hidden size For the RNN, varying embedding dimension does not have a lot of impact, but increasing hidden layer dimension does. For Transformer models, it is the opposite, as increasing embedding dimension is more important than varying hidden layer dimension (Figure 7.2, column 1).

Batch size vs learning rate We observe a similar behaviour between all models: the higher the learning rate, the higher the batch size must be (Figure 7.2, column 2).

7.2.2.2 Less significant parameters

Number of layers For both models, the number of layers does not change the performance, as long as the number of epochs is increased to reach convergence (Figure 7.2, column 3, 1st image).

Transformer heads number The head number does not have a statistically significant impact on the best Transformers' accuracy (Figure 7.2, column 3, 3rd image). However, for models with worse parameter combinations, using 4 heads increases instability and decreases convergence rate.

RNN attention type The dot attention outperforms the other attentions from Luong et al. (2015a), themselves outperforming no attention or attention from Bahdanau et al. (2015) (Figure 7.2, column 3, 2nd image).

7.2.2.3 Overall best parameters

Final parameters are described in Table 7.4, and, again, confirm our previous experiments.

Model	Learning rate	Batch size	Embed. dim	Hidden dim	#layers	Attention type/#heads
M-NMT^R	0.001	10	24	72	2	Luong-dot
M-NMT^T	0.005	30	24	72	4	3

Table 7.4: Results of parameter exploration experiments for RNN and Transformer models.

7.3 Main task results

Now that best performing models have been identified, we study their results, as well as how data augmentation methods can best be used to increase performance.

7.3.1 Baseline: bilingual setup

1-best Results At a first glance (Figure 7.3, "S" columns), SMT and RNN appear to have relatively similar results, varying between 58.1 and 76.9 BLEU depending on the language pair, outperforming the Transformer by 5 to 15 points on average. However, SMT performs better for IT⟷ES (pair with the least data), and RNNs for the other pairs. This confirms results from the literature indicating that SMT outperforms NMT when data is too scarce (Dowling et al. 2018; Singh and Hujon 2020; Skadiņa and Pinnis 2017), and seems to indicates that the data threshold at which NMT outperforms SMT (for our Romance cognates) is around 3,000 word pairs overall for RNNs, and has not been reached for Transformers.

n-best Results The BLEU scores for NMT and SMT increase by about the same amount for each new n ($n \leq 10$), reaching between 79.3 and 91.9 BLEU score at $n = 10$ for RNN and SMT. The Transformer, however, does not catch up.

7.3.2 Leveraging extra data

Pretraining, backtranslation Both pretraining the models and using backtranslation (Figure 7.3, "P" and "B" columns) increase the results of the Transformer models by 1 to 9 points, though they are still below the RNN baseline. It is likely the added monolingual data mitigates the effect of too scarce bilingual sets. The impact on RNN performance is negligible for most language pairs, apart from the lowest resourced one (ES–IT), for which backtranslation increases results. Lastly, these methods seem to mostly decrease SMT performance, likely because monolingual lexicons are noisy for our task, diluting the original (correct) bilingual data; this is less of a problem for NMT models, because they are then fine-tuned on the cognate task specifically. A detailed analysis of the backtranslated data can be found in Section 7.4.1.

Multilinguality Data augmentation through a multilingual setup (Figure 7.3, "M" columns) seems to be the most successful data augmentation method for RNNs (increasing performance almost all the times), and allows them to finally outperform bilingual SMT for the least-resourced pair as well (ES⟷IT). The Transformers benefit less from this technique than from adding extra monolingual data, apart for ES⟷IT, most likely for the same reason as earlier: this dataset being the smallest, adding words in ES and IT from other language pairs helps to learn the translation and stabilises learning. This technique is not applicable to SMT.

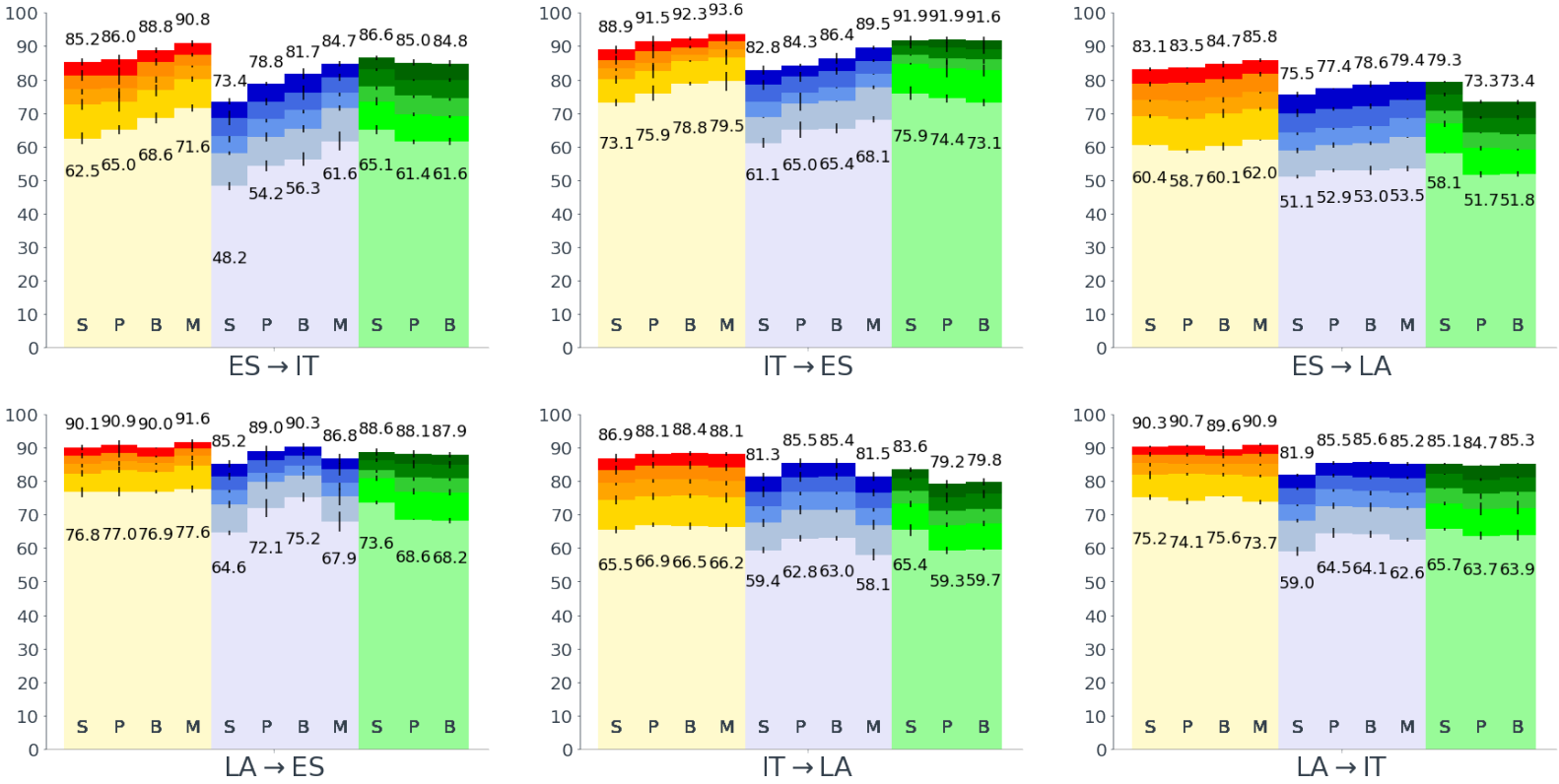


Figure 7.3: BLEU scores comparison on real Romance data. Colours indicate the model type: RNNs in orange (col 1 to 4), Transformers in blue (col 5 to 8), SMT in green (col 9 to 11). Colour shades indicate the value of n in n-best predictions (1, 2, 3, 5 and 10 from bottom to top). The letters (x-axis) indicate the setup: S - standard/bilingual, P - with pretraining, B - with backtranslation, M - multilingual.

7.3.3 Combining data augmentation methods

We choose to combine the best performing data augmentation technique overall, multilinguality, with pretraining and backtranslation (Figure 7.4) for our NMT models.

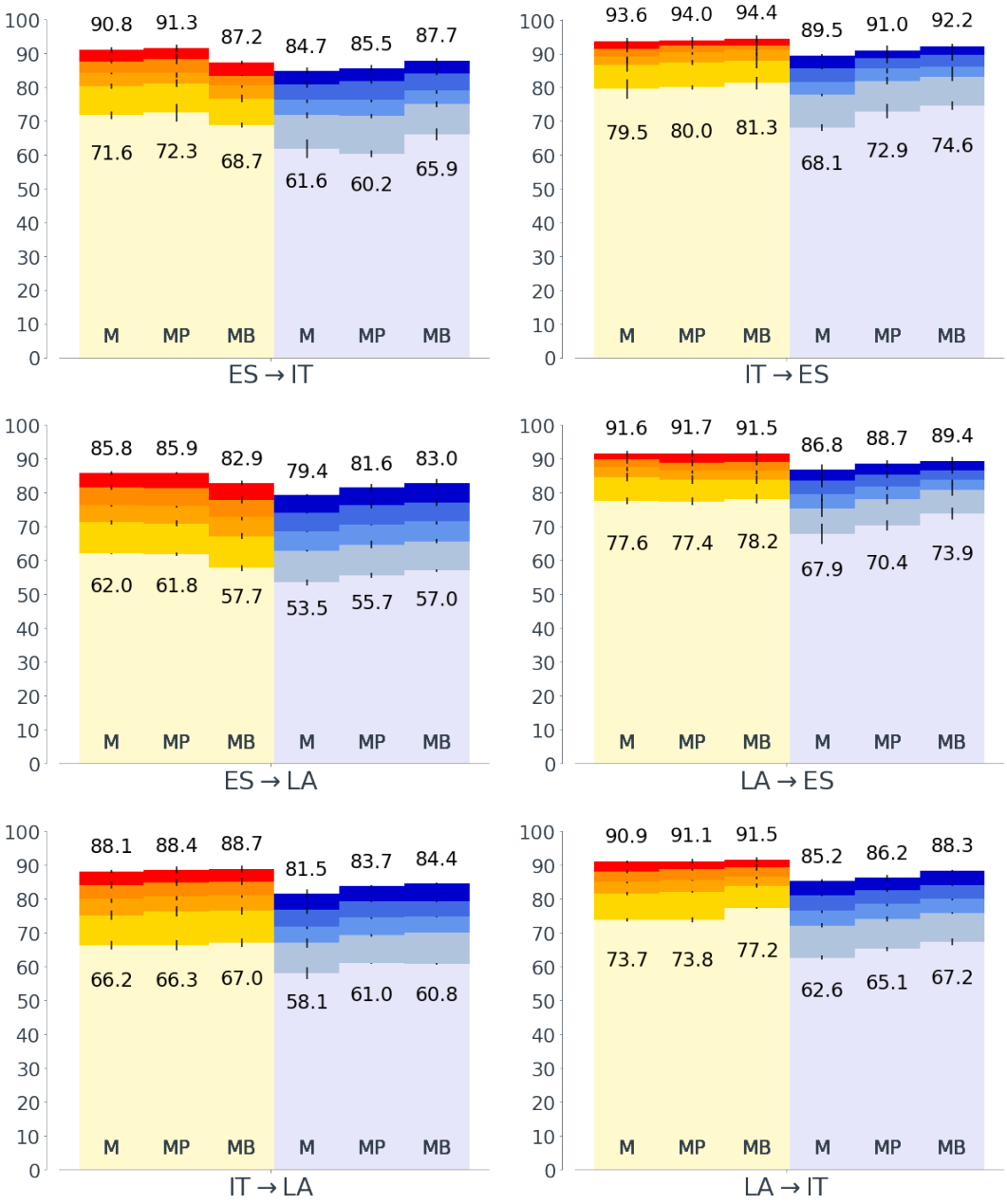


Figure 7.4: BLEU scores: RNNs in orange (col 1 to 3), Transformers in blue (col 4 to 6). Colour shades indicate the value of n in n-best predictions (1, 2, 3, 5 and 10 from bottom to top). On the x-axis, the letters indicate the setup: M - multilingual, MP - multilingual with pretraining, MB - multilingual with backtranslation.

Multilinguality + pretraining Combining multilinguality with pretraining has virtually no significant impact on the RNNs' results with respect to multilinguality only. For the Transformers, however, it increases the results by 2 to 3 BLEU on average.

Multilinguality + backtranslation Combining multilinguality with backtranslation provides the best results overall for Transformers (both being the best performing methods for these models). For the RNNs, however, the performance increase is smaller for most languages, and we even observe a decrease in performance when translating from ES (which was not the case with bilingual models).

7.4 Extended analysis

7.4.1 Remarks: observations of back-translated data

We study the quality of back-translated data, as not all new pairs of back-translated data were kept: if the word predicted was not also present in the corresponding monolingual lexicon, the word pair was discarded. Table 7.5 presents the number of word pairs kept after cleaning.

Final translation direction	IT→ES	ES→IT	IT→LA
Original #words	78,446 (ES)	99,012 (IT)	18,697 (LA)
Kept after B-NMT^R	21,808 ± 1,116	30,853 ± 740	17,507 ± 643
Kept after B-NMT^T	20,821 ± 1,182	32,560 ± 881	15,993 ± 1,023
Final translation direction	LA→IT	ES→LA	LA→ES
Original #words	99,012 (IT)	18,697 (LA)	78,446 (ES)
Kept after B-NMT^R	27,747 ± 1,754	14,238 ± 192	15,207 ± 1,155
Kept after B-NMT^T	21,606 ± 2,176	13,598 ± 1,092	14,110 ± 383

Table 7.5: Number of kept words pairs in the reversed parallel lexicons produced by the back-translation.

When looking at the number of word pairs produced by backtranslation for training, we observe that, after cleaning, we kept more than 65% of the words produced from LA monolingual data (for final translation directions into LA), against only 20% of the words produced to LA, and around 30% of the words from ES to IT and reverse (Table 7.5). Either producing words from LA to its daughter languages generated more plausible words because of a better model performance (least ambiguous translation direction), or because the smaller monolingual dataset contained less noise.

7.4.2 Choosing the best languages in a multilingual setup

Since multilinguality seems to have the most impact, we study the relevance of the language pairs used in the multilingual setup, we train additional multilingual neural models on only 1000 pairs of ES–IT data (single set), complemented by either nothing (to act as baseline), an extra 600 pairs of ES–IT, or 600 pairs of ES–L and IT–L (L being either Latin, a parent language, French,

a related language, or Portuguese, more closely related to Spanish than Italian). The rest of the data (Table 7.6) is split equally between dev and test.

Bilingual	FR–IT	FR–ES	PT–IT	PT–ES
#words	666	657	1,503	1,874
#phones	8,698	8,530	20,738	25,867
#Unique phones	40	43	36	41
Word length	6.53	6.49	6.90	6.90

Table 7.6: Supplementary bilingual lexicon statistics.

As we saw in section 7.3.1, the Transformers' scores are far more affected by low resource settings than the RNNs. We therefore study the impact of adding extra languages with RNNs only.

Baseline	ES→IT	IT→ES
1000 pairs	53.9 ± 3.4	66.6 ± 4.2
Added data	ES→IT	IT→ES
Same language pair	62.5 ± 2.5	71.8 ± 1.7
Latin	57.1 ± 1.8	67.4 ± 3.3
French	58.5 ± 2.0	67.0 ± 2.8
Portuguese	58.8 ± 1.1	66.9 ± 2.9

Table 7.7: BLEU for different multilingual settings.

Results on our new low-resourced baseline are lower than our previous baselines by around 10 points (Table 7.7), which is expected, since we use less data for training.

Adding 600 pairs of ES–IT words has more effect on ES–IT performance than adding any other pair of related languages, which indicates that, unsurprisingly, the best possible extra data to provide is in the language pair of interest. When adding a related extra language, the results are better than with the initial data only. From Spanish, the performance is best when adding Portuguese, its most closely related language, then French, then Latin. From Italian, we observe the opposite trend. Adding an extra language seems to help most to translate from, and not to, the language it is most closely related to. For very low-resource settings, where extra pairs of the languages of interest might not be available, it will probably be interesting to explore using extra languages related to the source language.

7.5 Linguistic analysis

We discuss the results of the best performing models for the best seed across all architectures (SMT, M-NMT^R with pretraining and M-NMT^T with backtranslation) from ES→IT. More than a third of the predicted words are above 90 BLEU[4] (resp. 35.4/46.4/38.1% for SMT/M-NMT^R/M-NMT^T), and for error analysis, we study the words below this threshold. The observations generalise to other language pairs.

[4]To study the BLEU of individual words, we use the sentenceBLEU function from `sacreBLEU` (Post 2018) with its default parameters.

Source context	Target context	Source	Target	SMT pred.	M-NMT^R pred.
(a) *Ambiguous sound correspondence not learned well*					
corvino 'raven'	*corvino* 'id.'	[kɔɾβino]	[korviːno]	[kɔrviːno]	[kɔrbiːno]
liebre 'hare'	*lepre* 'id.'	[lieβɾe]	[lɛːpre]	[liebrɛː]	[liɛːvre]
(b) *Form of the cognates changed too much*					
calaña 'kind/sort'	*quale* 'what/which'	[kalaɲa]	[kwaːle]	[kalaɲː]	[kalaɲː]
pie 'foot'	*piede* 'id.'	[pje]	[pjɛːde]	[pɛ]	[pɛːre]
(c) *Data error*					
suspirar 'to sigh'	*squillan* '(it) rings'	[suspiɾaɾ]	[skwilːan]	[sospiraːre]	[sospiraːre]
frenesí 'frenzy'	*frenetico* 'frenetic'	[fɾenesi]	[frenɛtiko]	[freːnezi]	[fronɛsː]
(d) *Model mistake*					
licencioso 'licencious'	*licenciozo* 'id'	[liθɛnθjoso]	[litʃentsioːzo]	[litʃɛntsioːzo]	[litʃɛntso]

Table 7.8: Prediction errors examples across ES→IT datasets for both SMT and M-NMT^R.

7.5.1 Predictions (word level)

Close Results We observe a lot of inaccurate but very close translations (e.g. Spanish *conveniente* 'convenient', phonetised [kɔmbenjɛnte], was predicted as corresponding to Italian [konvenjɛnte] instead of [konveniɛnte], with only one phone different, and coherently so). Sometimes these translations have a very bad score: Spanish *pulpito* 'pulpit', phonetised [pulpito], was predicted as [pʊlpiːto] instead of [pulpito], two close pronunciations, for a *sentenceBLEU* score of only 20.

Analysis of wrong results Wrongly predicted cognates correspond to four cases, as defined in Table 7.8.[5] We carried out a manual error analysis on all our predictions, and observed that their distribution was similar across models (respectively SMT/RNN/Transformer):

(a) 84.6/81.4/79.5% were cognates with an ambiguous sound correspondence (e.g. Spanish [β] to Italian [b/v/p]).
(b) 10.3/13.4/11.6% were cognates that had either evolved too far away from one another or contain rare sound correspondences, such as *pie* 'foot', phonetised [pje], predicted [pe] and [peːre] instead of [pjɛːde] *piede* 'foot'.
(c) 0.9/0.9/0.9% corresponded to data errors, such as *suspirar* 'to sigh', phonetised as [suspiɾaɾ], which was predicted as [sospiraːre] *sospirare* 'to sigh', its actual cognate, instead of its erroneous counterpart in our database ([skwilːan] *squillan* '(it) rings').
(d) 4.3/4.3/8.0% were model errors, such as "forgetting" part of a word during translation.

7.5.2 Usefulness of n-best results

We present here at which position the best prediction (according to sentenceBLEU, from `sacreBLEU` (Post 2018)) occurs amongst the 10-best predictions. For example, when going from Spanish *terroso* 'muddy', phonetised [tɛroso], to Italian *terroso* 'muddy', phonetised [terroːzo], the RNN predicted [teroːzo], [terɔːzo], [tɛrroːzo], [terroːzo], [tɛrrosː], [tɛrrɔːzo], [tɛrrɔsː], [terrɔːzo], [tɛrosː], and [tɛrosːo]: the correct result corresponds to the 4th position.

[5] Statistics are provided for the best models of the best seed, but examples are taken across seeds and models.

For all multilingual models, we computed the sentence BLEU score for each of the 10-best predictions and saved the position of the highest scoring prediction. We averaged these positions for all words in the test set and calculated the standard deviation. Table 7.9 contains the full results.

	IT→ES	ES→IT	IT→LA	LA→IT	ES→LA	LA→ES
SMT	1.08 ± 1.93	2.12 ± 2.55	2.01 ± 2.50	1.67 ± 2.30	2.49 ± 2.68	1.30 ± 2.14
M-NMT^R	1.04 ± 2.03	1.67 ± 2.42	2.20 ± 2.54	1.63 ± 2.38	2.51 ± 2.68	1.38 ± 2.30
M-NMT^T	1.34 ± 2.17	1.94 ± 2.34	2.42 ± 2.65	2.17 ± 2.57	2.78 ± 2.73	1.64 ± 2.31

Table 7.9: Average position of the closest prediction to the reference amongst the 10-best predictions.

The average position at which the best prediction (according to development BLEU) occurs (in 10-best predictions) is between 1 and 3 (Table 7.9). The lowest indices occur for Spanish (between 1 and 1.7) and Italian (between 1.6 and 2.2). The highest indices encountered occur when going for IT→LA or ES→LA (between 2 and 3). This illustrates the importance of n-best prediction when predicting cognates from child to parent languages, due to ambiguity. Standard deviations are between 2 and 3: for these languages, when studying cognate prediction, it is interesting to at least check the 5-best results.

7.5.3 Correlation between BLEU and confidence

	IT→ES	ES→IT	IT→LA	LA→IT	ES→LA	LA→ES
SMT	-0.02	0.05	0.10	0.06	0.11	0.05
M-NMT^R	0.26	0.43	0.23	0.29	0.47	0.28
M-NMT^T	0.21	0.39	0.20	0.26	0.38	0.28

Table 7.10: Correlation between model confidence and BLEU score.

When looking at the correspondence between model confidence level (evaluated as the log-likelihood of the prediction) and actual 1-best BLEU score for the predictions of the ES→IT set, we observe a Pearson correlation coefficient of 0.12 for the SMT, 0.47 for the Transformer, and 0.55 for the RNN. The model confidence is not correlated at all with the prediction accuracy for the SMT, and not that well for NMT (for other language pairs, the correlation coefficient is even lower, see Table 7.10). The NMT models tend to be overconfident, with an average confidence score of 81% for the RNN and 74% for the Transformer, and the SMT model underconfident, with an average confidence of 27%. A confidence over 70% for an actual BLEU score below 30 occurs in 26 cases for the RNN, 14 for the Transformer, and never for the SMT. The opposite (confidence < 30% and BLEU > 70) never occurs for the NMT, and occurs in 39 cases for the SMT.

7.6 Conclusion

We observed that optimized models seem to be able to learn historical word prediction in real-world setups, and not just our artificial data. This confirms that using machine translation architectures can benefit this task. Above a certain training data size, SMT and multilingual RNNs

provide the best BLEU scores for the task, SMT still being unrivalled when it comes to smaller datasets (which coincides with previous work comparing SMT and NMT for low-resource settings, and our previous analysis on artificial data, see Chapter 6).

When studying how to increase the amount of training data seen by our models, we found that exploiting the multilinguality of NMT architectures consistently provided better results than using extra monolingual lexicons (through pretraining or backtranslation), which contain noise for our task; combining the methods provided a significant amelioration for Transformers only. Leveraging multilinguality by training with extra languages also proved interesting, and we found the best possible extra data to add in a multilingual setting is, first, data from the languages at hand, followed by pairs between them and a parent language, then finally data from additional languages as close as possible to the source language.

We conclude that low-resource machine translation architectures and augmentation techniques are applicable to real historical word reconstruction as much as they were applicable to artificial data (as long as the task specificities, intrinsic ambiguity which requires n-best prediction and reliance on cognate data only are taken into account).

We now wonder whether it is possible to investigate what, precisely, our neural networks learn during cognate prediction. This is the focus of the next chapter, during which we will first extend the number of languages our models see, going from three to nine, then analyse in detail what our models learn, whether through external behavior or by designing specific probes to pinpoint properties of interest in the networks.

8 What can we learn from probing cognate prediction models?

> We ignore the blackness of outer space and pay attention to the stars, especially if they seem to order themselves into constellations.
>
> Stephenson (1995)

As we demonstrated earlier, historical word prediction can be, to an extent, modelled as a machine translation task. However, when using our multilingual encoder-decoders for cognate prediction, each encoder learns to map cognate phonetic data to an intermediate representation, common to all languages, and each decoder learns to map this common intermediate representation to its target language. What could this intermediate representation contain? As we hope that each encoder and each decoder learns the specific sound rules of its language, it would make sense that the best intermediary, between cognates of related language, and obtained through the sound rules of all, would be something akin to one of the many possible proto-forms, and this is what we will investigate in this chapter, first, by looking at the external behavior of the model, then by designing probes to understand its inner components.

8.1 Experimental setup

We will be optimizing our models for the cognate prediction task: generating, from a phonetised word, the plausible phonetic forms of its cognates in related languages.

Data We use our massively multilingual cognate dataset in 9 Romance languages: Galician (GL), Portuguese (PT), Spanish (ES), Catalan (CA), Occitan (OC), Italian (IT), French (FR), Romanian (RO) and Aromanian (RUP) (see Section 5.2.2). We use splits of 85/7.5/7.5% for the train/dev/test sets, using 3 different shufflings.

Models We use the SMT model as baseline. We compare all our possible NMT^R setups (bilingual, multilingual, without or with added monolingual data, sharing components or not) – encoders use a single-layer Bi-GRU (embedding dimension: 20, hidden dimension: 50), and de-

coders a single-layer GRU with Luong dot attention (embedding dimension: 20, hidden dimension: 50). Each neural model is trained using the Adam optimizer (learning rate: 0.005, batch size: 30) and the cross entropy loss, stopping on the first of either convergence of dev-BLEU or 15 epochs.

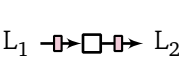
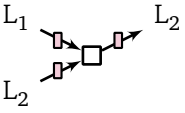
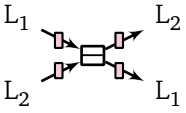
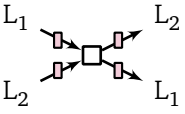
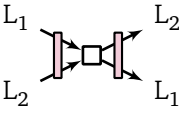
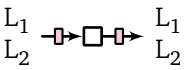
B-NMT	B-NMT+m	M-NMT
		
M-NMT+m	**M-NMT+m+shared_emb**	**M-NMT+m+shared_all**
		

Table 8.1: Model type reminder (see Section 5.1).

8.2 Steps of Analysis

In this chapter, we are not only interested in the raw performance of our models, but mostly in the results' interpretability. The core of this section is therefore not focused on the previous training objective, but on the following analyses and interpretability experiments.

Raw analysis We will first analyse our models and try to understand what they learned based only on their raw scores and prediction errors, as was done by Fourrier et al. (2021) and Meloni et al. (2021), to see the amount of linguistic information we can extract as such: which models perform the best (have the highest BLEU), and on which language pairs? What does it tell us about the languages we study? What can we learn about the different errors our models make when predicting on the test set?

Then, in order to compare the insights we got from a 'black box' analysis to insights obtained when looking specifically for linguistic or historical information, we design the following probing tasks.

Synchronic Probes Cognates are representative of their language phonetics, and we want to study whether the models learn deeper linguistic information while training on them.

- **Phonotactics**: To study whether our models learn phonotactics (the allowed arrangement of sounds and sound patterns in a language, see Section 1.1.1),[1] we adapt the ***bigram shift*** probing task (Conneau et al. 2018) to test whether encoders are sensitive to legal phone orders. A binary classifier is trained to distinguish between hidden representations of normal words and words whose phones have been inverted.

[1] Phonotactics, in a sense, is the 'syntax' of phonology.

- **Phonology**: To study whether our models learn phonologically meaningful representations, we study our vocabulary representations, as in Madsen et al. (2021). We reduce the dimensionality of our encoded phones representations using PCA (Pearson 1901) and t-SNE (Maaten and Hinton 2008) and look at the emerging underlying organisation of the phonetic space.

Diachronic Probes Cognates carry the historical information of the evolution of their respective languages. We want to see how much of this information was explicitly learned by the model.

- **Sound Correspondences and Contextualised Changes**: Cognates are usually identified by sound correspondence sets, which they also help define (see Section 1.2.1.1). We first study which correspondences the model has learnt to predict, to extract general statistics of sound classes prediction. We then use phones belonging to known sound rules, from Boyd-Bowman (1980), to study predictions out of context. Lastly, Meloni et al. (2021) provide sample sets containing minimal examples of artificial subwords in some Romance languages corresponding to a given sound correspondence and the associated Latin parent. To see if our models learn these sound correspondences, we study if they can reconstitute these sets.
- **Proto-form Reconstruction**: We go back to proto-form reconstruction to test our insight on hidden representation learning. Cognates all descend from a common ancestor word, their proto-form. To study whether the model learns historical information about said proto-form, we design a probing task where we train a decoder to predict a Latin word from the fixed encoded representation of its children Romance cognates.

8.3 Raw results

The full BLEU score tables of all our models on all our language pairs are in Appendix D.1, summarized in Figure 8.1, where we plot the heatmap of the BLEU scores for each language pair and each model, with high/low scores in red/blue, and big/small datasets indicated by +/—. We compare all our setups (SMT, B-NMT, B-NMT+m, M-NMT, M-NMT+m, M-NMT+m+shared_emb, M-NMT+m+shared_all, see Table 7.1).

8.3.1 General trends

When looking at SMT BLEU, we observe that, at a first glance, it seems that best results (60 BLEU upwards) are obtained for bigger datasets, and worse results (40 BLEU and less) for smaller datasets. This trend seems also exacerbarated for our B-NMT and B-NMT+m models, where this distinction becomes clearly visible, with smaller datasets at 10 BLEU and less for B-NMT and a bit more when adding monolingual data - they confirm precedent experiments on the impact of data size on BLEU. Furthermore, all the bilingual models also observe an obvious 100 BLEU for going from a language pair to itself. The trend is different for our multilingual models, where going from a language to itself does not always reach 100 BLEU, most likely because, since the intermediate representation is shared between all languages, it is optimized for none.

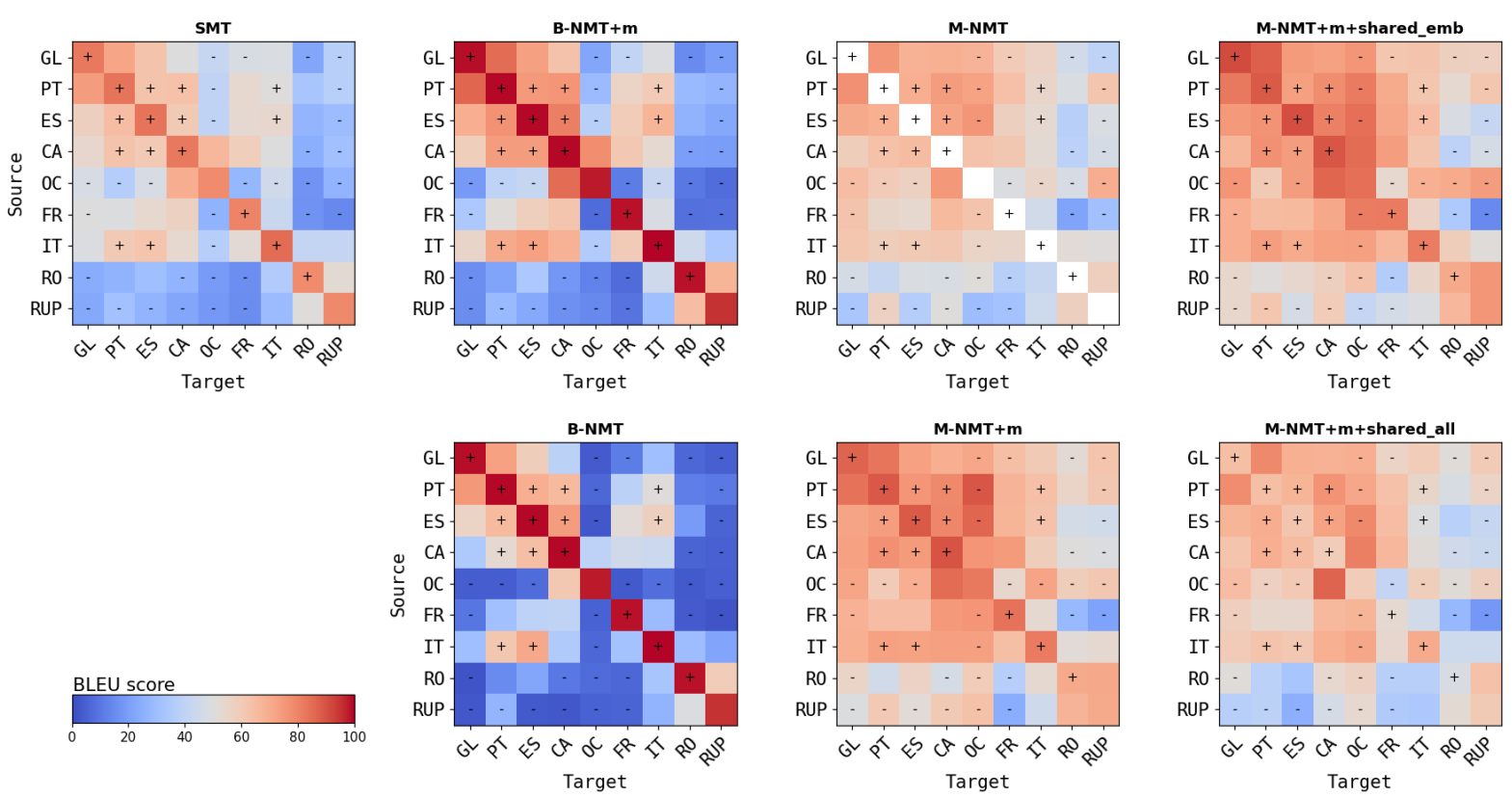


Figure 8.1: Heatmap of the BLEU scores for each model. Input languages on the vertical axis, and target languages on the horizontal axis. Data size is indicated by a "+" for more than 1000 word pairs, "—" for less than 300 word pairs. B for bilingual, M for multilingual, +m for added monolingual data, +shared_emb when sharing embeddings, +shared_all when sharing a single encoder and a single decoder across all languages.

In our M-NMT model, the BLEU scores stay the same or diminish slightly for the best results (compared to the bilingual models), but the least doted language pairs all see a considerable increase in performance. This increase can be due in part to the bigger quantity of data available, but it is likely not the only factor at play, otherwise our B-NMT+m model would outperform the M-NMT model, having had access to the biggest datasets we have (the monolingual sets).[2] We hypothesize that part of this increase could be due to learning transfer between our languages: for example, Portuguese and Spanish perform better to Occitan, with an almost 20 BLEU increase, which could come from transfer from the closely related Catalan, with more resources for our task, and especially between itself, PT and ES. In parallel, the decrease in performance for the highest resourced pairs could be because the encoders and decoders are now less specialised.

When adding monolingual data to our multilingual model (whether sharing components or not), the BLEU scores increase even more, but that not all language pairs are affected in the same way: PT, ES and FR to OC, or OC to RUP increase considerably more than, for example, RO and RUP to the other languages. It is likely that adding even more data (through monolingual information) increases the transfer between related languages (e.g. the Occitan monolingual dataset only contains 553 words, but could be helped by the added 2612 Catalan words).

8.3.2 Best Setups Choice

To choose best models, we synthesise the respective performance accuracies of our models in Figure 8.2, comparing their BLEU scores. This heatmap indicates the percentage of language pairs for which a model (left) is better than another model (bottom).[3] Both B-NMT models perform worst than the SMT baseline (with and without monolingual data). Multilinguality improves the performance, as the M-NMT model outperforms the baseline in 58% of cases. However, the best results are obtained when the models see the most data; the different M-NMT+m models outperform all other models for 80% of language pairs minimum. Another slight increase is obtained by sharing embeddings, as the M-NMT+m+shared_emb outperforms the M-NMT+m model in 58% of cases, but sharing encoders and decoders downgrades the performance. We will therefore focus on the M-NMT+m and M-NMT+m+shared_emb models, our two best setups.

8.3.3 Impact of language on performance

General results To study model performance on all language pairs separately, we focus on the previous heatmap of average BLEU scores (Fig 8.1) for our two best architectures (M-NMT+M and M-NMT+M+shared_emb) and the baseline (SMT). Our models and baseline behave similarly, with overall good BLEU scores, which seem to be slightly correlated with data size, except for some outliers.

[2] For French, for example, the monolingual dataset contains 3772 words, whereas the sum of all the others reaches only 3021 words, among which some might be redundant.

[3] Sums not equal to 100% indicate that the models have the same performance on some language pairs (ex: B-NMT and B-NMT+m).

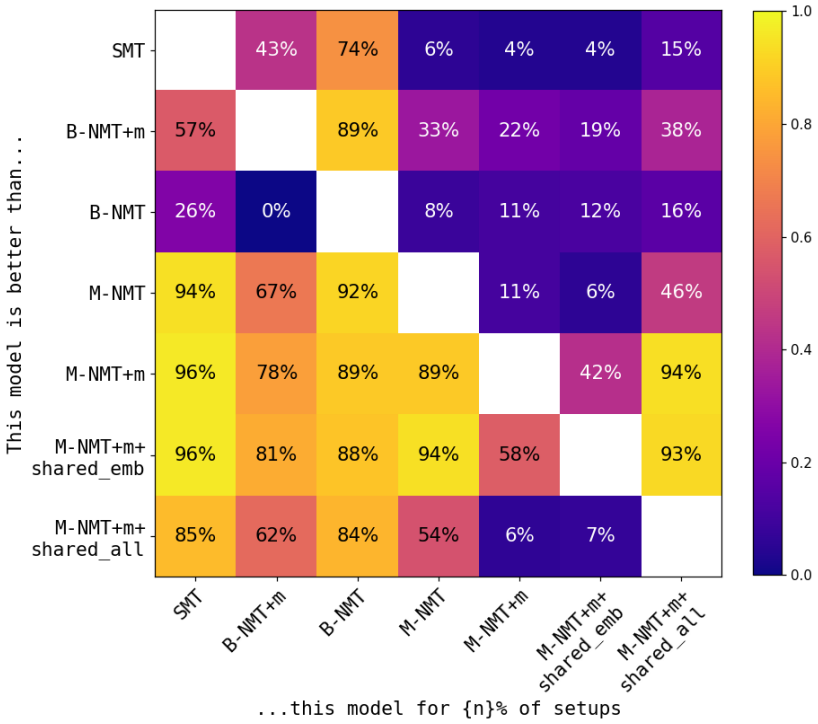


Figure 8.2: Percentage of language pairs for which a given model (left) outperforms another (bottom).

Outliers Firstly, predicting RO and RUP from/to all other languages has a considerably lower BLEU than all other pairs, except for RO–RUP itself: predicting between languages from too dissimilar language branches (Eastern-Romance and Italo-Western Romance), unsurprisingly, seems harder than translating within either of those branches. Secondly, GL⟷PT and OC⟷CA have higher BLEU than we could expect based on data size only.[4] In all setups, it therefore appears to be easier to predict cognates for closely related languages. This confirms experiments on multilingual language models, which show that transfer learning occur more easily for closely related languages sharing a script (Muller et al. 2021).

Heatmap symmetry We can also observe that all heatmaps are mostly symmetrical: it is approximately as easy to go from language 1 to language 2 than it is the other way around. Some clear exceptions to that rule are going from IT to PT and ES, which is always easier than the other way around, and, for our multilingual models, going from PT or ES to OC, easier than the opposite.

[4] It is important to note that this could also be linked to similarities introduced by our phonetisation method, as we used the Catalan phonetizer for Occitan and the Portuguese phonetizer for Galician.

8.3.4 Transfer learning hypothesis

In our Section 8.3, we hypothesized that part of the interest of multilingual models was the appearance of transfer learning between related languages, reinforced with added monolingual data. We highlighted this on Occitan, and we therefore test this hypothesis on predicting Occitan from Portuguese, and testing if Catalan helps. Catalan and Occitan are closely related, the foremost being 'high-resourced' for our setup, with a monolingual set of 2616 words, and the latter a 'low-resourced' language for our setup at 533 words in the monolingual set.

To check whether there is transfer from the high ressourced Catalan to Occitan, we compared the respective performances of several models on the prediction of Occitan: our baselines are B-NMT+m, M-NMT, and M-NMT+m Portuguese to Occitan, that we compare to using B-NMT+m Portuguese to Catalan, or M-NMT and M-NMT+m Catalan decoders in zero-shot to try to predict Occitan. The results are in Table 8.2.

Model	Trained on PT-CA	Trained on PT-OC
B-NMT	55.8 ± 8.8	6.7 ± 1.4
B-NMT+m	60.1 ± 12.8	28.1 ± 5.8
M-NMT	67.3 ± 12.1	67.1 ± 4.1
M-NMT+m	62.7 ± 6.1	86.1 ± 1.5

Table 8.2: Transfer learning experiments: we use several models trained on PT-CA or PT-OC to predict PT-OC.

We can first observe that using a B-NMT model originally trained on PT-CA (1031 word pairs) to predict PT-OC outperforms the model solely trained on PT-OC (223 word pairs) by 50 BLEU. This confirms that 1) we need to have more data pairs than a threshold (likely around 1000 word pairs) to learn anything using NMT models; 2) Catalan and Occitan are similar enough to expect transfer from one language to the other. Adding monolingual data augments BLEU by 5 points for the PT-CA model, and 20 for the PT-OC. This confirms our experiments on artificial data, specifically that we need at least 1000 word pairs to get a good BLEU: adding monolingual data only increases data size to 776 word pairs for PT-OC. Using M-NMT models benefits both to the Catalan and Occitan decoder for Occitan prediction: this confirms that there is a form of transfer inside the model, likely linked to the converging encoding space. Lastly, best results are obtained for the PT-OC M-NMT+m model (86 BLEU), whereas the PT-CA M-NMT+m sees its performance decrease. Adding monolingual data helps decoders to specialize on predicting their specific language. If we need zero-shot transfer learning later, it might be necessary to use models trained without monolingual data for the high-resource language to get best results.

8.3.5 Language relatedness

We also tried to see if it was possible to extrapolate language relatedness from the models' BLEU scores (by assuming that the closer a language pair, the better the BLEU). We perform agglomerative clustering of our languages, using BLEU scores as closeness metrics, then display the corre-

sponding dendrogram, using Virtanen et al. (2020), see Figure 8.3.[5] Of course, it must be noted that a dendrogram is an approximation of a much more complex historical ground truth (as a tree does not take into account areal influences, for example).

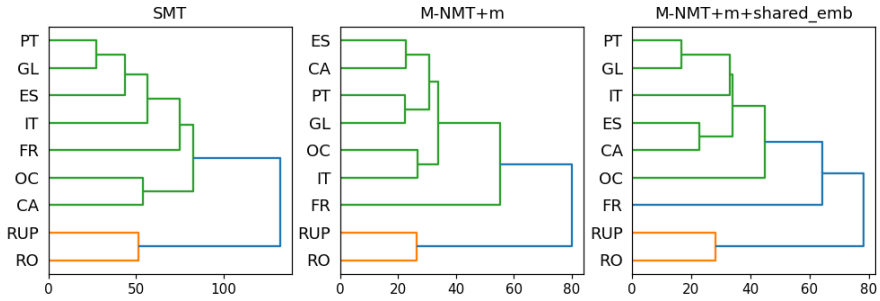


Figure 8.3: Generated dendograms from the language pair BLEU scores for our best performing models.

We observe that all dendograms successfully reconstruct the Eastern-Romance vs Italo-Western Romance division, as well as the Portuguese-Galician cluster. On the other hand, Italian is never the first branching in the Western-Romance subtree of the dendrogram. The SMT dendogram is the closest to the ground truth, with correct Ibero-Romance (ES, GL, PT) and Occitano-Romance (OC, CA) clusters, and French placed as closest to Occitano-Romance, while Italian is (incorrectly) grouped with the Ibero-Romance parent. Interestingly, multilingual dendograms separate French from the rest of the Italo-Western branch, which, as we saw in the chapter introduction (Section 5.2.2), makes sense, as its lexicon has undergone considerable Germanic influence. The M-NMT+m+shared_emb dendogram makes less sense from a linguistic point of view. On the other hand, the M-NMT+m dendogram groups Galician and Portuguese together (which is correct), as well as Occitan with Italian and Catalan with Spanish (which is not). However, Occitan and Catalan, though closely related, have undergone external influences, Spanish for Catalan, and French for Occitan. This small experiment seems to confirm that overall, SMT and M-NMT+m give results coherent with language history (when we see BLEU score as a closeness metric), while sharing embeddings changes language proximity as perceived by the models, likely because information is lost.

8.3.6 BLEU across languages

We plot the BLEU frequency histogram, for all our language pairs, and our 3 best models (See Figure 8.4 here and Figures D.2 and D.3 in Appendix). We first observe that all our models have similar BLEU frequencies overall - our models therefore have similar modeling abilities (except when using our least resourced languages pairs: e.g. anything from/to OC and RUP, where the M-NMT models have a higher number of words with good BLEU, most notably thanks to previously evoqued transfer learning). Then, we can observe that for most language pairs, the BLEU scores follow a bimodal distribution,[6] with one local maxima at the 90-100 BLEU interval, usually not

[5] This experiment's results are also likely impacted by the data size, and would have been more conclusive with an equal number of data points for all languages. However, this would have meant reducing the size of datasets for all our language pairs to the size of the least doted one, Occitan to Aromanian, with only 81 words pairs, which is likely to be too small an amount to learn on for our neural networks.

[6] On some languages, we could say that it is trimodal, with a high peak at 0-10, another one at 90-100, and a flattened one in the middle.

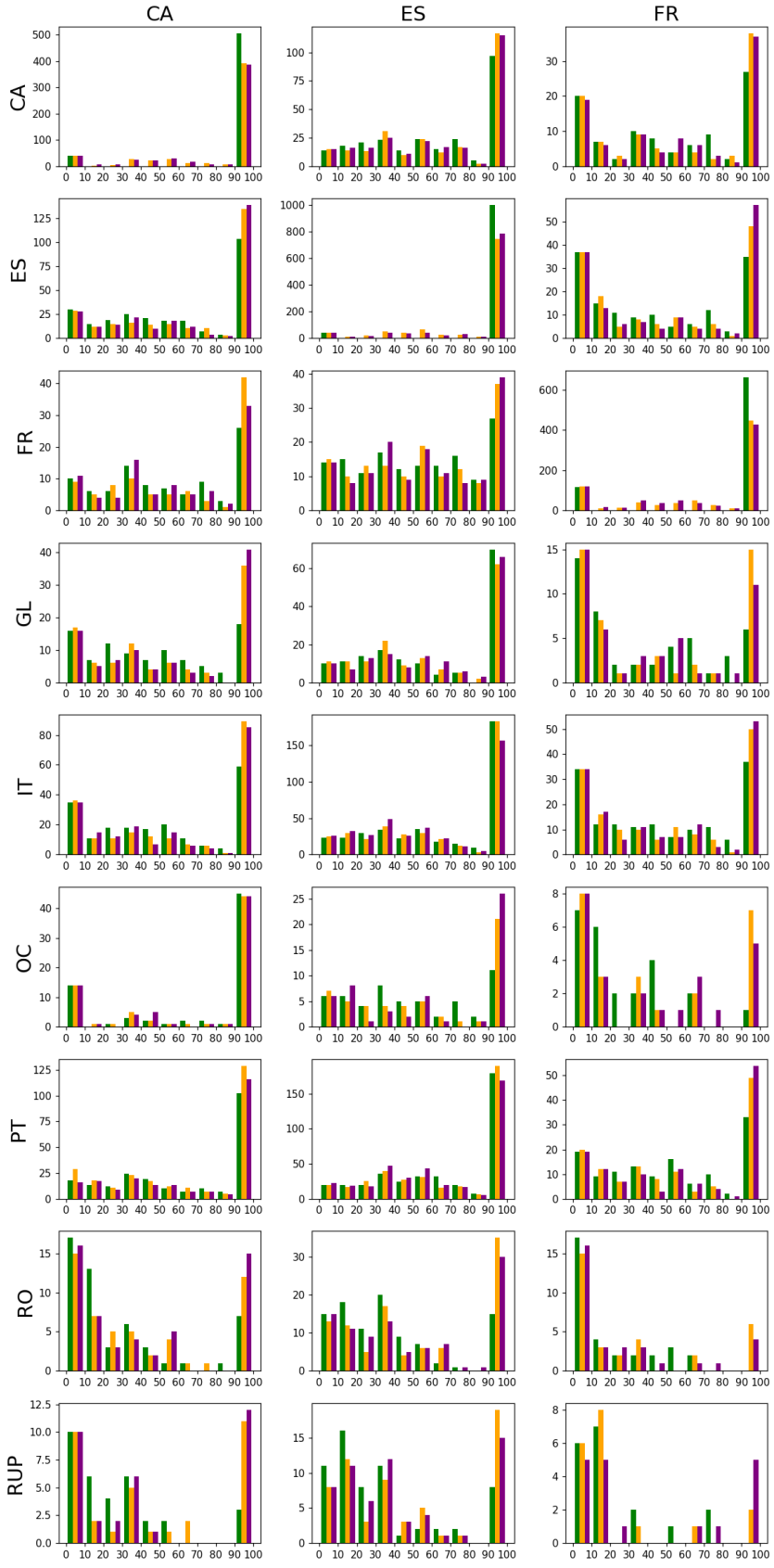


Figure 8.4: BLEU frequencies, from all our languages to Catalan, Spanish, and French. SMT in green, M-NMT+m in orange, M-NMT+m+shared_emb in purple.

spread out, and, most of the time, a skewed normal or uniform distribution in the lower values, with a mean in the 0-20 BLEU interval and a standard deviation above 20 BLEU. Each model therefore encounters, for a given language pair, 1) a non negligeable amount of words which are very easy to translate and 2) other words which, following linguistic evolution diversity in complexity, have BLEU scores more widely distributed, depending on the relatedness and data size of the language pair. Further studies could look at the differences between these two groups in terms of data history.

8.3.7 Conclusion

Analysing cognate prediction models by solely focusing on BLEU scores already allows us to draw conclusions on the nature of information learned by our models: cognate prediction is easier for closely related languages in a massively multilingual setup, and multilinguality allows the emergence of transfer learning from our most to least resourced language pairs. We also observe that not all words are as hard to predict, as almost all language pairs encounter cognates extremely easy to predict. Further analysis could be done on the linguistic difference between easy and hard cognates, but we focus on predictions of easier or harder language pairs.

8.4 Predictions (phone level)

We compare the predictions and errors made by the models in three cases: the language pair is highly resourced and gets a good BLEU score (ES-PT), the language pair has average resources but contains close languages and gets a good BLEU score (PT-GL), the language pair has almost no resource and gets a bad BLEU (RO-FR).

We use the Needleman and Wunsch (1970) dynamic programming algorithm as modified by Gotoh (1982) (adding affine gap penalties) to compute the pairwise alignment between our models predictions and the targets.[7] From these alignments, we extract correspondences of characters, in 1 or 2-grams.[8] We can then better see when the predicted phones match the gold, and study the models' outputs.[9]

8.4.1 Prediction categories

When looking at the phone level model predictions, we observe that they can be: (1) correct (equal to gold); (2) phonetically close to the gold (ex: [β], a voiced bilabial fricative, instead of [b], a voiced bilabial plosive); (3) other, which can either correspond to a known sound correspondence, incorrect in the current example but attested in others (ex: [v], a voiced labiodental fricative, instead of [b], a voiced bilabial plosive) or be a wrong prediction (ex: [a], a vowel, instead of [b], a consonant) (Table 8.3). In 2-gram, this classification becomes (1) correct (identical 2-grams); (2) close (identical/close phone and close phone); (3) other (the rest, which can then

[7] We use the implementation from Cock et al. (2009).

[8] Using 3-grams alignments provided no further insights.

[9] To remove noise which might be caused by incorrect alignments, we only keep correspondences occurring more than once, and in 2-grams, we discard the pairs which contained a blank inserted during the alignment process.

Pair	ES→PT			PT→GL			RO→FR		
Prediction	Correct	Close	Wrong	Correct	Close	Wrong	Correct	Close	Wrong
1-gram									
SMT	90.9%	5.3%	3.8%	95.5%	2.4%	2.1%	62.7%	12.7%	24.5%
M-NMT+m	89.1%	5.5%	5.4%	93.6%	3.4%	3.1%	71.6%	8.8%	19.6%
+shared_emb	90.7%	5.1%	4.3%	92.4%	3.9%	3.7%	73.8%	14.6%	11.7%
2-gram									
SMT	83.4%	9.7%	6.9%	93.2%	4.1%	2.6%	49.1%	14.0%	36.8%
M-NMT+m	81.5%	9.8%	8.6%	89.3%	6.2%	4.5%	64.4%	8.5%	27.1%
+shared_emb	83.3%	9.6%	7.1%	88.0%	6.8%	5.2%	58.6%	24.1%	17.2%

Table 8.3: Prediction types frequency for 1 and 2-grams, for three language pairs: ES→PT (good BLEU, big data size), PT→GL (good BLEU, average data size, close languages), RO→FR (bad BLEU, small data size).

be divided in (a) 'one correct/close and one wrong', or (b) 'two wrong' phones, other patterns almost not occurring).

We observe that in easy situations (lots of data, ES→PT, or language proximity, PT→GL), the statistical is slightly more accurate that the multilingual model. When the situation is harder (RO→FR), the part of "close" predictions (with respect to "wrong" predictions) increases considerably for the neural models, and the best performing one becomes the multilingual neural model, very likely because it can leverage information from the other pairs.

8.4.2 1-gram analysis

When looking at the wrong predictions in more detail, we observed that errors can either occur frequently or only one time:

- **ES–PT**: 20% of errors occur more than once. Most frequent errors (frequency > 6) for models are correspondences between [ɔ]/[u], [v]/[b], [ɛ]/[ɨ] or [i], so either known phonetic correspondences or vowels close on the vocalic triangle. The least frequent errors tend to be nonsensical.
- **PT–GL**: 17% of errors occur more than once, and they contain the same patterns (correspondences between vowels, or [b]/[v] and [k]/[ʒ]).
- **RO-FR**: Only 4% of errors appear more than one time, and almost all errors occur only once and are nonsensical.

When plotting phone accuracy between prediction and gold of character as a function of phone frequency for our three cases of interest (Fig D.4 in Appendix), we observe that for our big dataset, the BLEU is correlated to the character frequency count, which is not the case for the cases with smaller datasets: for our close languages, the BLEU is good no matter the character to predict (maybe because the characters to predict are easier), and for further languages, the results are bad all the time (but the frequencies of occurrence are low for all characters).

8.4.3 2-gram analysis

We divide our predictions in three groups: correct (identical 2-grams), close (one identical phone and one close, or two close), incorrect (the rest). We also single out the cases where both characters are close or both characters are wrong (parenthesis in the columns of Table 8.3). For our analysis, we discarded the pairs which matched a character with a blank symbol (inserted during the alignment process).

First, we can observe that the behaviour of the models varies: for language pairs with good BLEU scores, the ratio of "close results" to "wrong results" is between 1:1 and 2:1, whereas when the BLEU score is bad, this ratio reaches between 1:5 and 1:3. It is likely that language pairs with good BLEU learn to identify correspondence patterns correctly, but are mistaken in ambiguous situation (close results), when language pairs with bad BLEU do not even learn correct correspondences and spew random phones.

The close results are divided in "one correct, one close" and "two close" predictions. The model almost never predicts "two close" predictions (number in brackets). The wrong results are divided in "one correct or close, one wrong" and "two wrong" predictions (number in brackets). The model almost never predicts two wrong results for language pairs with enough data, but this ratio increases considerably for our language pair with little data.

When studying frequent cases (more than one occurrence) of "one correct/close and one wrong" errors for ES–PT, we observe that they are either a wrong vowel with the correct consonant ([ɔɾ]/[uɾ]), a correct phone with a [b]/[v] confusion, or less frequently, a correct/close vowel with a [k]/[ʒ], [w]/[l] confusion, or lastly, for the bilingual models only, the change between [ŋv] and [mb]. Almost no "one correct/close and one wrong" error occurs more than once for the other datasets.

The most interesting errors appear in our last category, when both characters are wrong. These errors never occur more than once for a letter combination, a model and a language pair, and a large quantity of them are nonsensical, or likely the result of alignment problems. However, some of them, for ES–PT, can be grouped in a single category (SMT/Bilingual NMT/Multilingual NMT: 30%/5%/30%): cases of methathesis, where the phones have been inverted (for example, [ɪŋ]/[nɨ] or [eɾ]/[ɹɨ]).

8.4.4 3-grams analysis

When studying 3 grams (not displayed in the table), we observed that almost no 3-grams are completely wrong (except for RUP–FR, where they are nonsensical) - they either have one or two approximately correct characters: there will be no insight available from studying 3-grams compared to 2-grams.

8.4.5 Conclusion

When looking at errors with a high apparition frequency,[10] which tend to be plausible and similar between neural models and baseline, we observed that wrong phones in 1-gram or 2-gram case (a) correspond to high-mid vocalic alternations patterns, ([ɔ]/[u], [ɛ]/[ɨ]-[i]), exchange of consonants linked by a sound correspondence ([v]/[b]), or less frequently, in 2-gram only, to a [k]/[ʒ] or [w]/[l] confusion.[11] 2-gram case (b) correspond to metathesis (phone inversions, ex: [ɪŋ]/[nɨ] or [eɾ]/[ɾɨ]) 30% of the time, the rest being nonsensical errors.

These results seem to confirm the observations made by Meloni et al. (2021): most errors made by the models are not arbitrary but tend to correlate with historical linguistic phenomena. We now try to investigate if those linguistic aspects also appear inside our models.

8.5 Synchronic Probing

8.5.1 Phonotactics

Probe Training We trained MLP classifiers to detect whether encoded words contain a switched bigram of phones or not. For a given language, the encoder used is either randomly initialised or coming from our multilingual models and frozen. This experiment is reproduced for all data shuffles and all languages. No matter the setup, the classifier performance is systematically around 50%, therefore no better than random.[12]

Fine-tuning We decide to try fine-tuning our multilingual models on the classification of bigram switches, to see if this is information our models can learn to distinguish: we use the same setup as for the probing tasks, except that the encoders are now fine-tuned along the classifier training instead of frozen. The results are again no better than random.

Conclusion When learning to predict cognates, the encoder does not spontaneously encode phonotactics information, nor does it learn to encode it when fine-tuned specifically on that. This is interesting, because sound correspondences relations between cognates are partly linked to phonotactics. If the model does not learn this information explicitly, it has to learn something else instead.

[10] Therefore not studying RO→FR, whose errors tend to occur only once and be nonsensical (likely the result of the difficulty of learning on so little data).

[11] SMT also produce a segment voicing change between [ŋv] and [mb].

[12] Even accounting for the fact that this task could be made harder by the fact that some switched bigrams are phonotactically allowed, we should still observe a better accuracy than random.

8.5.2 Phonetics

Following Madsen et al. (2021), we study the hidden representation of our neural models. We encode every available phone in our languages of interest and use two dimension reductions techniques (PCA, t-SNE) to look for patterns.

8.5.2.1 Vowels

When plotting the 3-dimensional PCA for the vowels' hidden representation, we observe two patterns of spatial organisation (Figure 8.5).

Language Relatedness Along one dimension, the space seems to be organised through a linguistic continuum (with vowels in French together, then the rest of the Gallo-Romance branch, then the Eastern-Romance branch, then the Ibero-Romance branch).[13] However, this ordering is not constant across data shuffling; depending on the data seed, the model places different languages close to one another in the intermediate representation—models learn a language separation of the space, but not any constant language phylogeny.

Phonetic Organisation Along the other two dimensions, we observe a pattern of phonetic organisation seemingly similar to the vocalic diagram (which is obtained when organising vowels along their production height and tongue advancement, see Section 1.1.1) and this pattern proves stable; all our NMT models, no matter the data shuffling trained on, seem to have the three vocalic poles in their PCA ('u/o', 'i/e', and 'a'), more or less some outliers. (We color phones according to their theoretical vocalic pole on the vocalic diagram.)

Looking deeper, these outliers fall in two categories (Figure 8.5): rare French phones (nasal vowels for example, which do not exist in the other Romance languages, and therefore are harder to place) or phones that have actually been clustered with the most similar pole orthographically, and not phonetically. For example, [ɔ] is linked to 'u/o' instead of 'a' (and both [ɔ] and [o] sounds usually come from the letter o), [ɛ] is linked to 'i/e' instead of 'a' ([ɛ] and [e] from e). The model seems to have learnt to encode similarly phones occurring in similar contexts, and not phones that are actually phonetically similar. However, 1) phones occurring in similar contexts in our cognates usually come from the same original sounds, and therefore tend to be phonetically similar and 2) phones occuring in similar contexts are very interesting for historical linguistics. We can therefore say that, though the models ***seem*** to have learned a 'phonologically meaningful taxonomy of phonemes without explicit supervision' (Meloni et al. 2021), a faithful and not just plausible interpretation (Jacovi and Goldberg 2020) is that they have actually learned something akin to a 'phonetic language model,' where phones encoded similarly are first contextually similar.

[13] Clustering phones on their respective languages is the main feature we observe when using t-SNE.

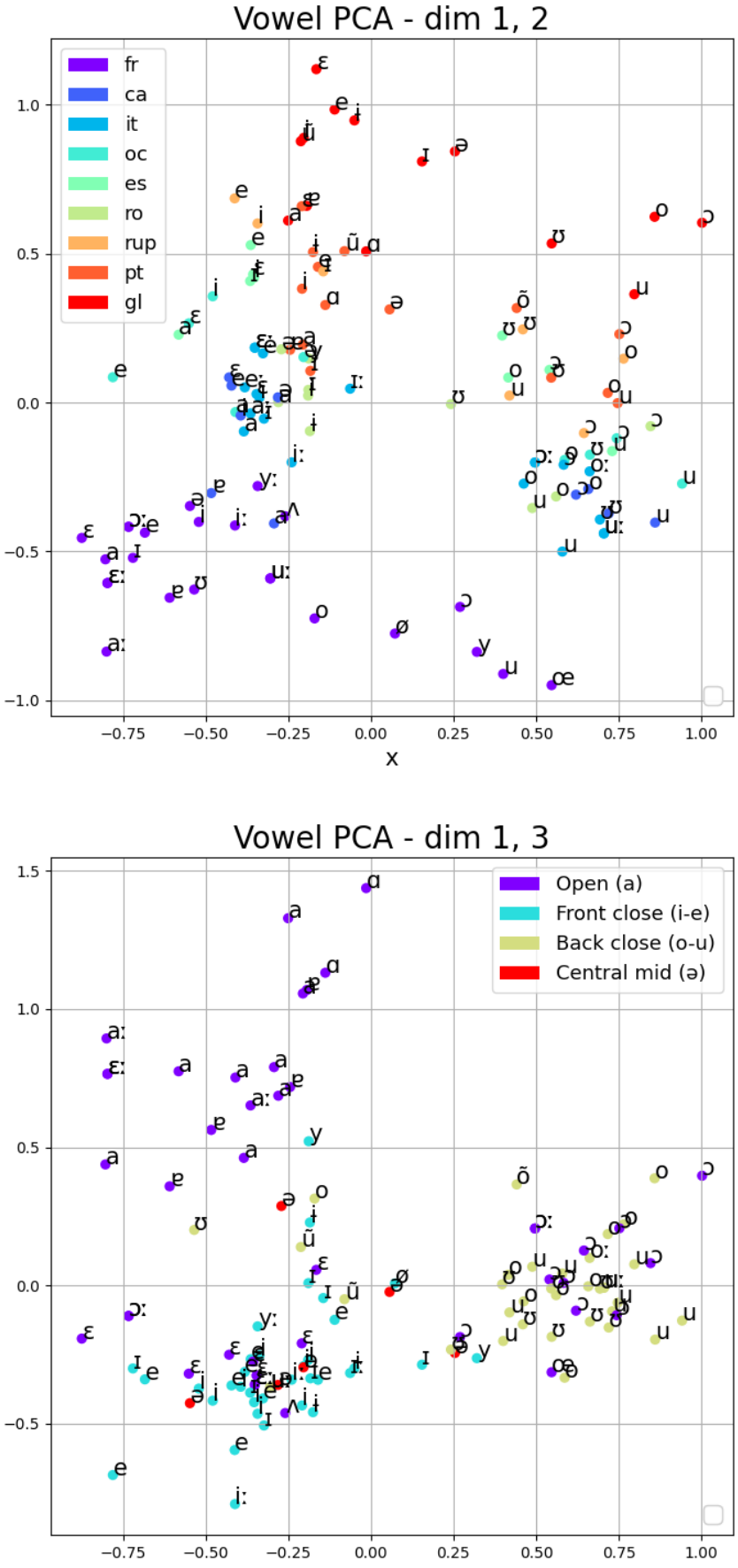


Figure 8.5: Vowels PCA, seed 0. Top coloured on language. Bottom: coloured on pole of the vocalic triangle.

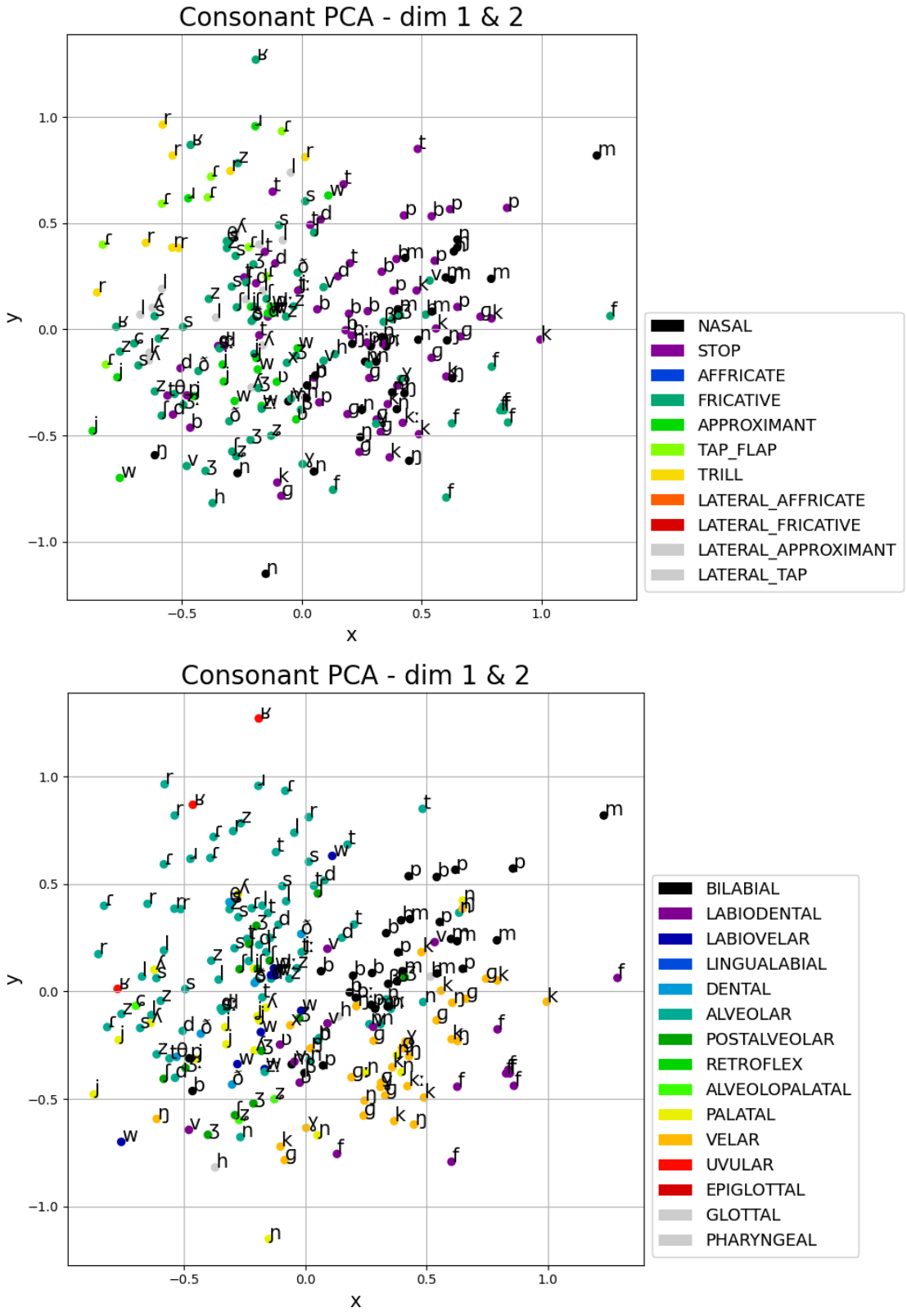


Figure 8.6: Consonant PCA, seed 0, coloured on manner above and on place below.

8.5.2.2 Consonants

We plot the PCA (Figure 8.6) and t-SNE (Figure D.1 in Appendix) for consonants, coloured on either manner or place, and observe the same patterns. Letters seem to be grouped phonetically at a first glance, but are actually grouped by orthographic context more than phonetic similarity: ([b], [β], [v] together, or [g], [ɣ], [k] together, and so forth).

8.6 Diachronic probing

8.6.1 Do the models learn phone correspondences?

In this section, we focus on French, Italian, Spanish and Portuguese, as well as Romanian when we have the data. Boyd-Bowman (1980) compiled a list of contextualised sound correspondences between the first four languages, as well as their Latin origin; we extended this set with vowels data from Wikipedia. Meloni et al. (2021) described minimal sound sets between Latin and several daughters (the five forementioned languages). We successively use these datasets to test our models' learning.

8.6.1.1 General phone class statistics

We look at general learning statistics for phone classes:[14] are some harder to predict than others?

For each language pair studied, we ask our models to predict 5 best equivalents of each possible phone (using a single phone as source word), and kept the answers with a confidence above 1%. We then compare our results with statistics extracted from our training set, where we align source and target words at the phone level, using the Needleman and Wunsch (1970) alignment algorithm (edited with a custom cost function to bring phonetically close sound together, see Appendix D.5.6).

When plotting precision and recall averaged across languages, for all phones, depending on phone class (Figure 8.7), we first observe that both our M-NMT+m models (with or without shared embeddings) outperform SMT in terms of both precision and recall, except for approximants. We also observe that M-NMT+m has better recall, but M-NMT+m+shared_emb better precision on average. Therefore, we can say that the latter, sharing embeddings across all languages, which forces the single embedding layer to have access to all phones, seems to help the model predict more relevant sound correspondences. On the other hand, using an embedding layer per encoder, which specialises each embeddings to their respective languages (M-NMT+m), seems to help the model remember more phones correspondences.

We also observe that the models have similar performance trajectories, as they all are better (in terms of recall and prediction) at predicting trills (above 85% recall and 75% precision for all models), then nasals, than they are for stops, vowels, or fricatives (between 90% and 60%

[14] Phone classes: vowel, nasal, stop, fricative, approximant, tap/flap, trill, or lateral approximant phones

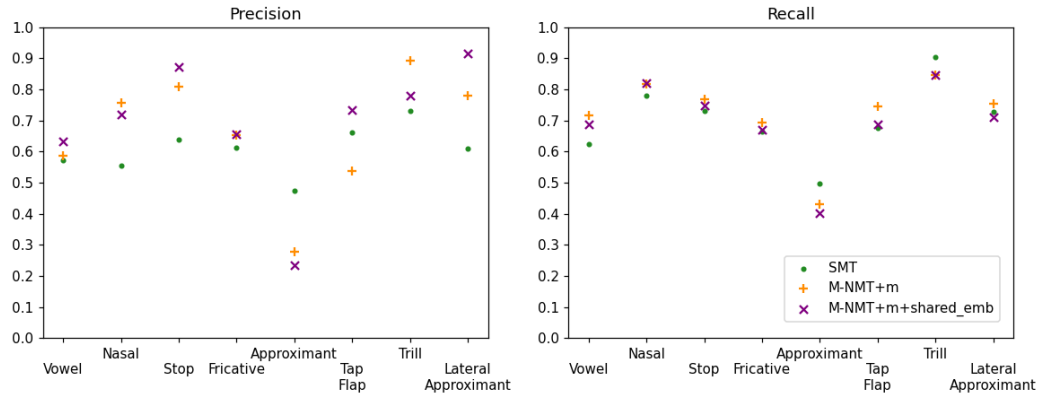


Figure 8.7: Average precision and recall for a given phone type, across all languages, for each model.

recall for 80% to 50% precision across models), the worst category being approximants (40 to 50% recall, and 25% precision for neural models against 50% precision for statistical models). To better understand these results, we first compute the percentage of each category in our datasets across languages (see Table 8.4).

Language	Approximant	Fricative	Lateral Approximant	Nasal	Stop	Tap-Flap	Trill	Vowel
Spanish	5.9%	26.5%	5.9%	11.8%	17.6%	2.9%	2.9%	26.5%
Italian	4.2%	18.8%	4.2%	8.3%	25.0%	2.1%	2.1%	35.4%
Portuguese	7.1%	16.7%	4.8%	9.5%	14.3%	2.4%	2.4%	42.9%
French	4.2%	20.8%	2.1%	8.3%	12.5%	0.0%	0.0%	52.1%
Romanian	6.5%	22.6%	3.2%	9.7%	19.4%	3.2%	3.2%	32.3%

Table 8.4: Phone type frequency in our dataset.

We observe that frequency tendencies are coherent across languages, with the most frequent categories being vowels and fricatives, then stops, nasals, approximants, lateral approximants, lastly tap-flaps and trills. The two categories for which results are the most homogeneous across model types are the most frequent categories. However, there seem to be no further correlation between frequency and precision or recall. To go deeper in our analysis, we therefore study the detailed correspondences between input phone type and output phone type, language pair per language pair, for all models against the baseline, in a more comprehensive figure.

At a first glance, approximants tend to be aligned with a wider variety of phones and phone types, where some other categories, such as stops, tend to correspond to stops in other languages (See Section D.5 in Appendix). To observe these alignments in more detail, we plot these correspondences in a more comprehensive figure (Figure 8.8) - the global rows represent a common source language, the columns a common target language, and in a given cell, each input phone category is associated to 4 columns: output from SMT, M-NMT+m, M-NMT+m+shared_emb, as well as the gold output, framed in black. In general, the models seem to behave similarly: for all languages, vowels are mostly corresponding to vowels, nasals to nasals, stops to stops, approximants to vowels, fricatives or approximants, and lateral approximants to lateral approximants or fricatives. Interesting correspondence appear for the tap/flap and trill, which do not exist in all target languages (absent in the gold column).

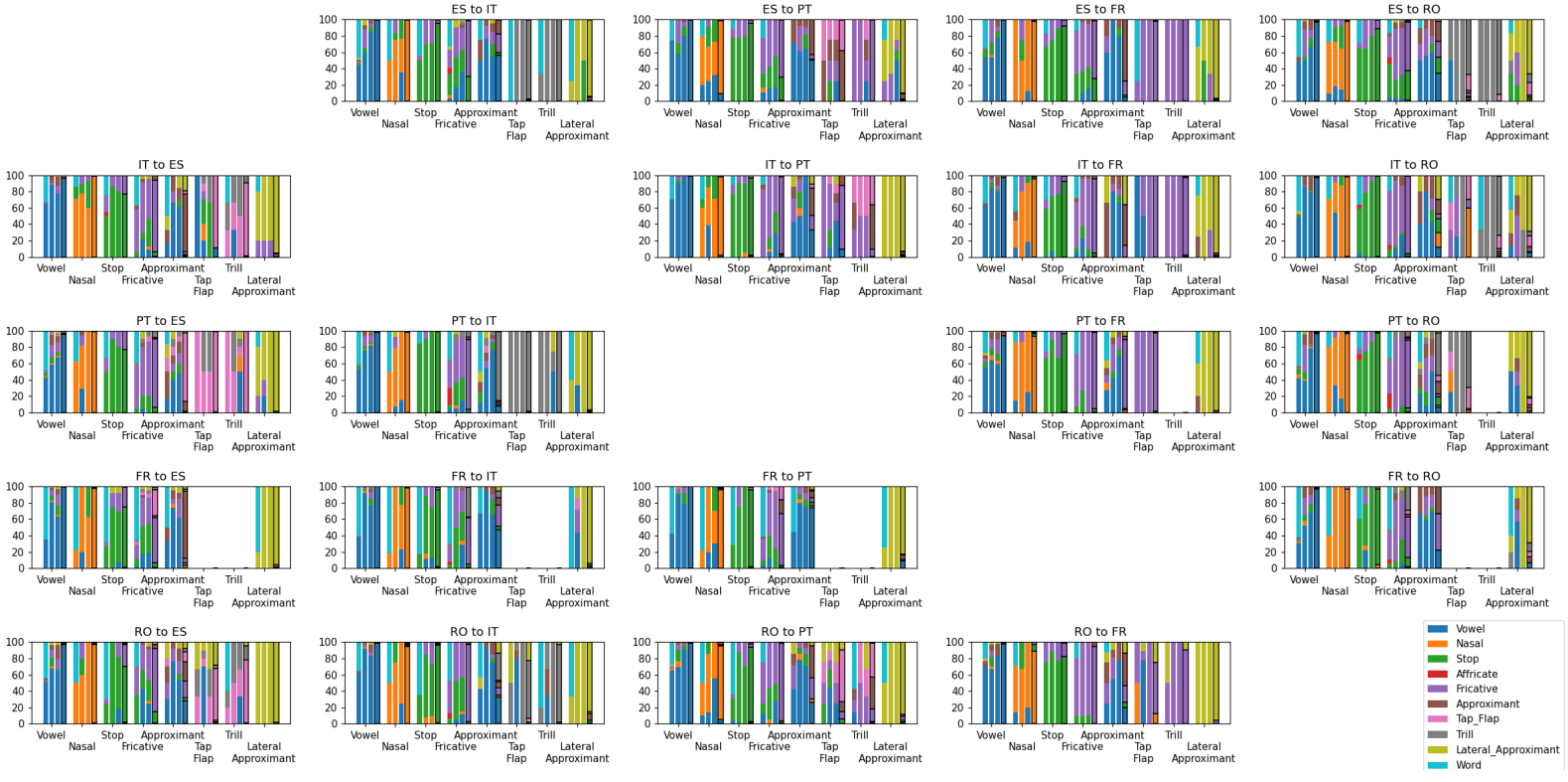


Figure 8.8: Phone type predicted for each model and language pair.

Grid rows are the source languages, columns the target languages. For each cell, columns are grouped by 4 depending on phone type in input (vowel, nasal, stop, fricative, approximant, tap/flap, trill, lateral approximant). The first 3 columns of each set contain the SMT, M-NMT+m, M-NMT+m+shared_emb phone type outputs for the current given input phone type. The last column represent the target gold phone types for our data.

From Portuguese, they are both linked to tap/flaps to Spanish, trills to Portuguese, and fricatives to French. We also observe that the baseline for some categories is more varied across languages than for others: approximants in input, for example, often correspond to 3 to 6 possible and phone types for model outputs, when the other phone types usually correspond only to one or two significant outputs. This can explain the troubles we had earlier with approximants precision and recall.

We look specifically at the performance of each model individually. SMT performance seems bad at a first glance, because SMT seems to consistently output word-like predictions instead of single phones: SMT is penalised because it has not seen words so short during training.

The multi-phones predictions sometimes make direct sense ([θ] as [tʃ]), sometimes are just a variation on possible syllables with a reference phone ([d] is translated as [da], [di], [do]), but sometimes are nonsensical subwords ([oe] as [uno]).

The M-NMT+m and M-NMT+m+shared_emb are mostly coherent, with the main categories predicted similar to the baseline, except for approximants, and sometimes tap-flaps (from Italian). There seems to be a learning of sorts of what constitutes a 'phone class', though this could also be linked to the orthographic mapping we exposed in the PCA. We therefore want to see if there is a learned notion of sound correspondences in our models.

8.6.1.2 Uncontextualized phones correspondences

We now focus on the comparison between our results and sound correspondences extracted from the literature. Boyd-Bowman (1980) lists the sound changes between Latin and 4 descendants: Spanish, Italian, French, and Portuguese. From these sound changes lists, we extracted a sound correspondence list focusing on consonants of interest : p, b, v, realised phonetically as one of [p], [pː], [b], [bː], [β], [v], and often occurring in similar contexts, and t, d, realised phonetically as one of [d], [dː], [ð], [t], [tː] (Table 8.5). The following four, p, b, t and d, occur in all our languages of interest. We do not consider vowel correspondences, as vowels are less stable through time than consonants, and tend therefore to be harder to predict, appearing in less regular sound correspondences.[15]

Since sound changes lists are usually studying the most interesting relations, and not all, our sound correspondence list is not exhaustive, but it should be good enough to compare between the models' learning processes. We want to compare the precision and recall of the different models depending on their confidence: we look at the predictions which have a confidence score above 0.01 (we fit the 10-best probability scores of MOSES to sum to 1).

[15] All our correspondences can be found at `github.com/clefourrier/selected-romance-sound-correspondences`, and the file format is explained in Appendix D.5.7, with an example.

Phone	Source	Spanish Gold	Spanish SMT	Spanish M-NMT+m	Spanish +shared_emb	Italian Gold	Italian SMT	Italian M-NMT+m	Italian +shared_emb	Portuguese Gold	Portuguese SMT	Portuguese M-NMT+m	Portuguese +shared_emb	French Gold	French SMT	French M-NMT+m	French +shared_emb
p	Spanish					p: p	p	p	p	p	p	p	p	p	p	p	p
	Italian	β b p	p	p	p					b p	p pi	p	p	v p	p pa ip	p k	p
	Portuguese	b p	p	p	p	p: b p	p pr	p	p					b p	p pa	p	p
	French	p	pa p	p k ɾ	p	p: p	po p	p	p	p	pʊ p pjʊ	p ʃ	p				
b	Spanish					b: v b p	ve v b	v b f	v b b:	β v b p	v b	v	v b	b v f	v b	b v f	v b
	Italian	b β	b β	b p	b					β v b p	b vi v	b	b v	v b	b	b	b v
	Portuguese	b β	b β al	b p	b β p	b: v b p	b p	b p	b p					v b	b	b p	b v
	French	b	bo βo b	β b	b β a	b: b	bo b	b v	b a k	b p	bʊ bjʊ b	v b	b p				
β	Spanish					b: v b p	v b p	v b f	v b b:	v b β	b v vi	v b	b v k	v f	b v f	b p v	v b
	Italian																
	Portuguese	b β				b: b								v			
	French																
t	Spanish					t t:	t	t	t	ð t d	t	t	t	t	t	t	t
	Italian	ð t d	t	t	t					ð t d	t dʃ	t	t	t	t	t s d	t
	Portuguese	d tʃ t tɕ ð	t	t	t	t t:	t	t t:	t					i t it	t kt	t	t
	French	ð t d	to te tjo	t	t	t t:	to te ta	t t:	t	ð t d	tʊ ti t	t	t				
d	Spanish					t d d:	do de te	d	d f	ð t d	di	d	d f	t d	d	d	d
	Italian	ð d	d	d t j	d					d ð	d	d t	d	d	d di	d ʒ	d
	Portuguese	ð t d	d	d b	d	t d d:	d t di	d v	d					t d	d t	d	d
	French	d	d do ðo	d	d	d d:	do dio dia	d	d f	d	dʊ di dɐ	d	d				
ð	Spanish					t d	d di t	d v	d f t	ð t d	d	d	d f	t	d t	d	d
	Italian																
	Portuguese	ð t d				t d								t			
	French		to ta	t k	d t		to	t t:	d f i								
v	Spanish																
	Italian	b β	b β	b β f	b					v b	v	v	v b	v f	v f de	v ʒ f	v
	Portuguese	b β	b be ben	b f	b	b: v b	ve v ven	v f	v b					v f	v f b	f v b	v
	French	b β	b βo βe	b	b	b: v b p	ve vo v	v	v f b:	v b β	vɐ vi v	v	v				
p:	Italian	p	p	p	p					p	p	p	p	p	p	p k	p
b:	Italian	b β	p β β l	b p	b β f					v b β	b	b v p	b v	v b	b v	b v ʒ	b v
t:	Italian	tɕ tʃ t	t tʃ kt	t k	t					jt wt t	t	t	t	i t it	kt t pt	t s ʃ	t
d:	Italian	j d	bð	d b	d p t					ʒ d	d	d v t	t d p	d			

Table 8.5: Phone correspondences for our consonants of interest (p, b, v, d, t), extracted from Boyd-Bowman (1980) — for predictions, we keep 3-best with confidence above 0.01.

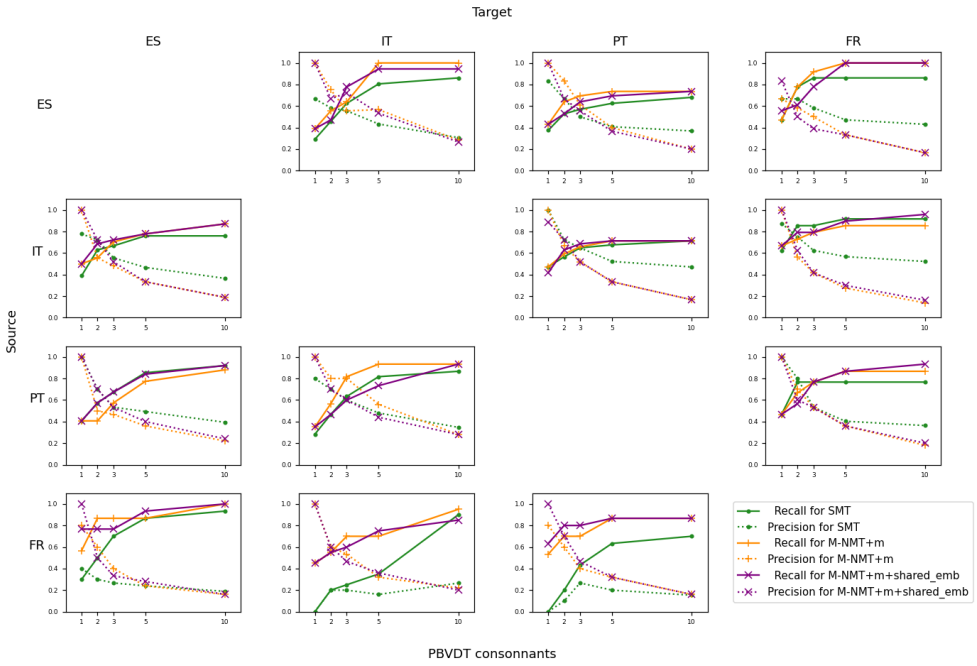


Figure 8.9: Precision and recall of our models for the consonants of interest (p, b, v, d, t), in 1/2/3/5/10-best.

General observations As we can see on Figure 8.9, when studying the percentage of sound correspondences correctly captured in 1, 3, 5 and 10-best for our different models and language pairs, the models perform quite similarly in 1-best, for a given language pair, from Spanish, Italian and Portuguese, and that the SMT model performs worse when predicting from French. An interesting point is that the M-NMT+m+shared_emb has most of the time an equivalent or better performance than the M-NMT+m in 1-best, when statistical models tend to be equivalent or worse. We also observe that, above a number of n-best predictions (usually 3, at times 5), we reach a plateau for the recall, when the precision keeps dropping. The plateau possibly occurs because 1) we only keep n-best results above a certain confidence score, to prevent random predictions, and it is very likely that the 6 to 10-best are actually not kept because they fall below the 1% confidence score; 2) consonants contain between 1 and 4 correspondences, and reaching 100% recall below 4-best would therefore be nonsensical.

Differences between the models On 1-best, the models are equivalent, with SMT being slightly worse overall. We can also observe that in most cases, our best model (M-NMT+m+shared_emb) reaches 80% recall or above from 5-best for consonants. When looking at the predicted results in more detail (Figure 8.5), we observe that SMT is prone to predicting more than one phone from a single phone (for example, [do], [dio], [dia], [da], [de], [dine] when translating [d] for French→Italian), which the neural models never do (which explains the lower precision of the statistical model, which, having almost never seen singular phones in its training sets, does not know how to handle them).

Spanish to	Italian	Portuguese	French	Romanian
SMT	1.5	2.4	1.4	1.8
M-NMT+m	1.5	2.0	2.0	1.7
+shared_emb	1.2	1.4	2.0	1.8
Italian to	Spanish	Portuguese	French	Romanian
SMT	1.4	2.7	1.4	2.0
M-NMT+m	1.4	2.1	2.6	1.7
+shared_emb	1.7	2.0	1.9	1.9
Portuguese to	Spanish	Italian	French	Romanian
SMT	1.4	1.6	1.1	2.5
M-NMT+m	1.7	1.9	1.7	1.4
+shared_emb	1.5	1.9	2.3	1.7
French to	Spanish	Italian	Portuguese	Romanian
SMT	1.4	2.8	3.2	1.6
M-NMT+m	1.5	1.9	1.2	1.9
+shared_emb	1.6	1.9	2.0	2.2
Romanian to	Spanish	Italian	Portuguese	French
SMT	2.7	2.6	3.5	2.3
M-NMT+m	1.3	2.1	1.4	2.0
+shared_emb	1.2	1.9	1.7	1.8

Table 8.6: Average position of the correct result in 5-best for the (Meloni et al. 2021) sound correspondences.

Spanish to	Italian	Portuguese	French	Romanian	Avg.
SMT	**76**	**73**	**64**	**73**	**71**
M-NMT+m	67	61	52	61	60
+shared_emb	61	61	58	64	61
Italian to	Spanish	Portuguese	French	Romanian	Avg.
SMT	**88**	64	**73**	**76**	**75**
M-NMT+m	61	**70**	27	58	54
+shared_emb	70	61	52	55	59
Portuguese to	Spanish	Italian	French	Romanian	Avg.
SMT	**88**	**82**	67	**76**	**78**
M-NMT+m	76	76	**76**	70	74
+shared_emb	73	67	55	67	65
French to	Spanish	Italian	Portuguese	Romanian	Avg.
SMT	61	67	36	**64**	57
M-NMT+m	70	**70**	**76**	61	**69**
+shared_emb	**73**	64	**76**	48	65
Romanian to	Spanish	Italian	Portuguese	French	Avg.
SMT	**72**	62	59	**62**	**64**
M-NMT+m	56	**69**	**66**	34	56
+shared_emb	53	**69**	62	41	56

Table 8.7: % of cases where our models predicted the good artificial correspondence among the 5-best predictions for the Meloni sound correspondences (Meloni et al. 2021). Best results in bold.

8.6.1.3 Phones in known sound correspondence relations

Meloni et al. (2021) provide sets of minimal phonemes test sequences representing known sound correspondences in Romanian, French, Italian, Spanish and Portuguese, to evaluate their models' generalisation. For example, the minimal set for sound changes linked to word initial Latin /pl/ is, for an artificial Latin origin [pla]: Romanian [pla], French [pla], Italian [pja], Spanish [ʎa] and Portuguese [ʃa]. They use it to reconstruct the Latin protoform from its children, but we use it to study whether our model can reconstruct the plausible correspondences between the daughter languages. We predict 5-best 'cognates' for the provided artificial segments, to see if our models can generalise sound correspondences too, as well as compute the average position for the correct result among the 5-best predictions (Table 8.6). We first observe that all models have similar behaviours: when answers are correctly predicted, they usually are predicted in first or second position on average (the neural models being better than the baseline for our linguistically more original languages, Romanian and French).

Our neural models reach between 54% and 74% average accuracy from a given language (Table 8.7),[16] and the statistical baseline tends to perform better overall. However, sound correspondences where the source languages are the most divergent in our Romance family (French and Romanian, see Section 5.2.2) are better captured with the neural models by 3 to 40 points (for language pairs with enough data, such as French→Spanish, Italian, Portuguese, or Romanian→Italian, Portuguese). Adding shared embeddings increases performance with our more typical Romance languages as source and decreases performance for the previous languages, while still performing better than the baseline. We can therefore say that sound correspondences information is captured by our models.

8.6.2 Do the models capture diachronic information?

We used very small RNN decoders with attention[17] as probes, and trained them to predict Latin proto-forms from the NMT encoded hidden representations of several models (Table 8.8).

These probes were trained to predict from either:

- Pretrained source-to-source **B-NMT+m** frozen encoders, which have learnt a coherent hidden representation of the source language, but possess no extra linguistic information
- Pretrained **M-NMT+m** frozen encoders, to assess if multilinguality is helpful in capturing latent historical information, and performing better of not than B-NMT+m encoders
- As **top baseline**: a B-NMT encoder trained along with the probe on the task of learning Latin from the current source, therefore a model specialized on learning Latin with an optimal hidden representation for it.
- As **bottom baseline**: a randomly initialized B-NMT encoder, to make sure that our probes

[16] We did not expect our models to reach a 100% accuracy, as the provided examples are minimal for a set, and not necessarily a sound pair between languages (some sounds could also appear in other sound correspondences), but reach nonetheless a comparable accuracy to Meloni et al. (2021) on their similar proto-form prediction task. However, in their paper, they teach their models to go from cognates sets to the proto-form, when we go from one cognate to a parallel cognate, and therefore work on a less ambiguous task, but from less information.

[17] Embed./Hidden sizes: 10/20, Luong dot attention.

are not too expressive (Conneau et al. 2018; Zhang and Bowman 2018). Too expressive networks can learn to fit any random noise, and have therefore no value as probes.[18]

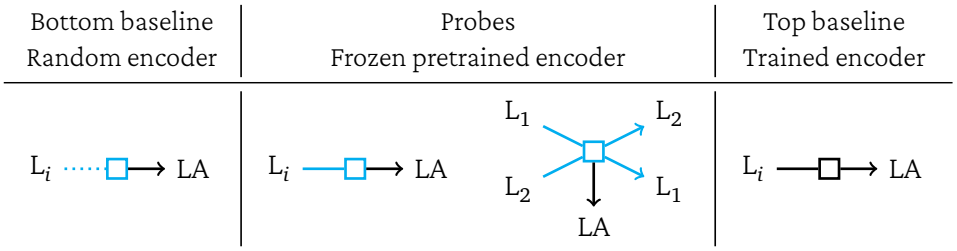


Table 8.8: Different probes used to study the presence of diachronic information in our models

On Table 8.9 and Figure 8.10, we plotted the BLEU test scores obtained at each epoch by the different setups for the different languages. First, performance with Occitan is considerably low, but data size of LA-OC is one order of magnitude smaller than the other sets. Then, for the rest of the languages, baselines behave quite differently. Our random baseline has low performance, which confirms that our probes are selective enough to prevent rote memorisation of anything from an random encoded representation. B-NMT+m and M-NMT+m encoders, without any fine-tuning on the prediction of Latin, reach or surpass the performance of our top probe, a model specifically trained on this task. The better performance of our multilingual model with respect to the bilingual one seems to indicate that there is something in the multilingual intermediate representation better helps reconstruct the proto-form.[19]

Model	Catalan	Spanish	French
Data size	2,276	4,332	2,511
Top baseline	32.3 ± 4.7	**46.7** ± 0.6	**31.7** ± 3.6
M-NMT+m	**36.8** ± 1.3	38.8 ± 2.4	**31.7** ± 0.9
B-NMT+m	28.5 ± 3.7	38.0 ± 1.9	29.9 ± 0.8
Untrained baseline	5.2 ± 0.9	3.1 ± 0.5	3.1 ± 1.0
Model	**Galician**	**Italian**	**Occitan**
Data size	1,167	5,606	399
Top baseline	23.8 ± 4.3	**50.5** ± 3.0	6.5 ± 1.0
M-NMT+m	**26.8** ± 1.9	45.1 ± 0.6	**9.6** ± 1.4
B-NMT+m	20.7 ± 2.1	44.0 ± 0.6	9.0 ± 3.1
Untrained baseline	2.8 ± 0.5	5.5 ± 1.8	1.8 ± 0.1
Model	**Portuguese**	**Romanian**	**Aromanian**
Data size	3,609	1,040	725
Top baseline	**36.4** ± 2.9	18.2 ± 6.2	9.9 ± 1.9
M-NMT+m	35.1 ± 0.6	21.1 ± 2.5	**18.1** ± 4.5
B-NMT+m	31.1 ± 0.9	**26.2** ± 0.8	16.8 ± 0.4
Untrained baseline	4.8 ± 0.7	2.6 ± 0.9	2.5 ± 0.3

Table 8.9: Probe BLEU test scores for 3 seeds (20 epochs).

[18] "As long as a representation is a lossless encoding, a sufficiently expressive probe with enough training data can learn any task on top of it." (Hewitt and Liang 2019)

[19] The M-NMT+m+shared_emb encoders reach half the performance of the M-NMT+m model: sharing embeddings seems to capture considerably less diachronic information, possibly because the phonetic information of all languages are mashed together.

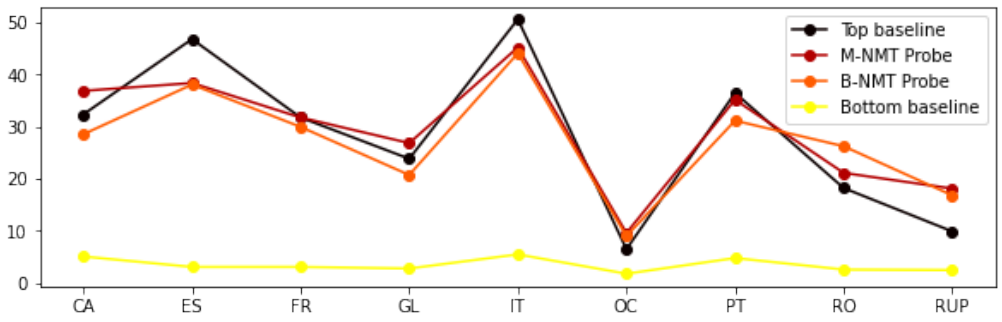


Figure 8.10: Average BLEU score for each input language and each probe setup

8.7 Conclusion

After training and selecting the best multilingual machine translation models for the task of cognate prediction, we confirmed the black-box analysis previously made of similar models (they capture language relatedness information and phonetic similarity).

We then probed our models, to find what precisely they learn, and discovered that, linguistically, they do not encode phonotactics, though they encode a form of phonetic similarity by grouping phonemes by similar contexts. We also discovered that our models learn diachronic information: they are able to produce sound correspondences, and, even more interestingly, they contain enough historical linguistic information to allow the reconstruction of the proto-form with no fine-tuning, performing as well as models trained specifically for this task. We can therefore conclude that, although the latent diachronic information is not present in a form we expected (phonetics, phonotactics), it is nonetheless present.

Further work is needed to understand more precisely under which form this information is stored, and what precisely it could tell us about our task and data.

Part III

EXTENSION

9 From cognates to words

> Fate does not take sides. It is fair-minded and generally prefers to maintain some balance between the likelihood of success and failure in all our endeavors.
>
> Towles (2016).

We saw in the previous part how low-resource character-level neural machine translation tools and techniques can be used for historical word prediction in related languages, as well as the modalities of such applications. In this chapter, we circle back to translation, by applying similar techniques to lexicon induction in low-resource situations. Pseudo-parallel lexicons have been successfully used to improve machine translation models, from managing out-of-vocabulary words in sentences through direct correspondence (Che et al. 2020; Li et al. 2016; Luong et al. 2015b) to more complex methods, such as fine-tuning pre-trained out-of-domain models to in-domain (Hu et al. 2019) or aligning word embedding spaces to generate language agnostic (or at least multilingual) embeddings (Artetxe et al. 2017; Duong et al. 2016; Mikolov et al. 2013b). Induced plausible lexicons have also been used to help field linguists find new words in very low-resourced languages (Bodt et al. 2018; Bodt and List 2019). This task therefore constitutes a broader application of our work so far, and a number of our previous assumptions might need to be reconsidered, as, at a first glance, we will switch from using phonetically related historical words to using orthographic lexicons of not necessarily related words.

9.1 Research

9.1.1 Linguistic context

For this task, we choose to focus on a French regional language, Alsatian. Alsatian is a West Germanic, High German, Alemannic German language. It is the second most spoken regional French language,[1] with 650 000 bilingual locutors[2] as of 2013, according to a report by the French Delegation for the French language and languages of France (Paumier et al. 2013), and presents a number of challenges. First, it is quite **low-resourced**, though initiatives have seen the light to try to generate more data in Alsatian, notably through the RESTAURE[3] project (for resource gen-

[1] The first being Occitan.

[2] Alsatian speakers live in a state of bilingualism, with French, the national language, having become the majority (when not unique) language of Alsace since 1970 (Huck et al. 2007).

[3] RESsources informatisées et Traitement AUtomatique pour les langues REgionales – Computational Resources and Processing for Regional Languages

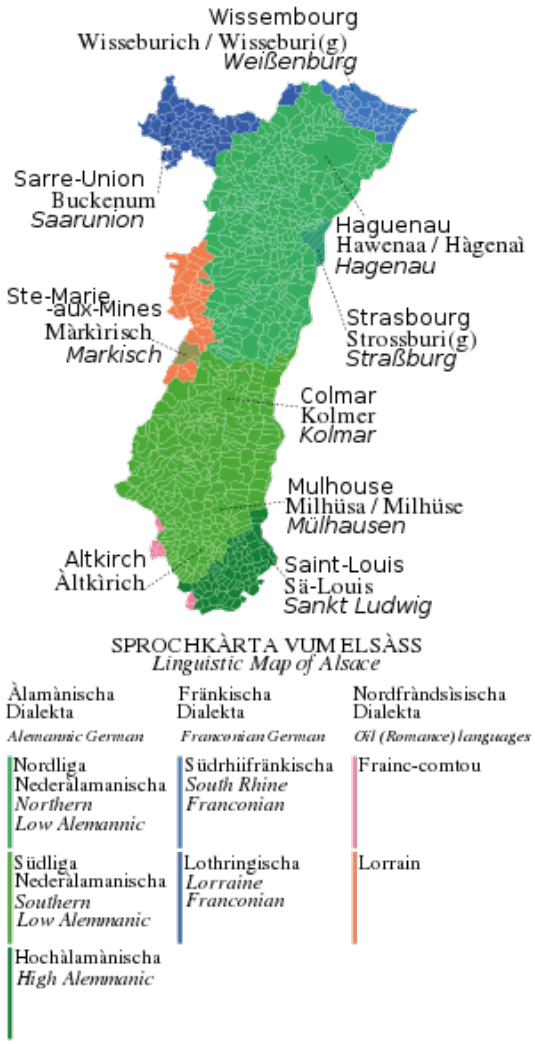


Figure 9.1: Linguistic Map of Alsace (CC BY-SA 2020 Nat/Wikipedia).

eration for regional languages). Available data contains lexicons generated through crowdsourcing (Millour 2020), automatic alignement (Bernhard 2014; Bernhard and Steiblé 2015), or the work of language conservation associations, such as the OLCA (Office pour la Langue et les Cultures d'Alsace et de Moselle),[4] or more specialised linguistic annotated data for part-of-speech tagging (Bernhard et al. 2018a; Millour et al. 2020), Named Entity Recognition (Bernhard et al. 2018b) and even a theatre corpus (Ruiz Fabo et al. 2020). Despite these initiatives, the quantity of parallel data publicly available is still likely too low for automatic machine translation, and automatic extraction of data is made noisy by the proximity of Alsatian to Swiss German, another Alemannic language sharing the same language tag (GSW) but 'among the most vital ones in Europe in terms of social acceptance and media exposure' (Scherrer and Rambow 2010a) and therefore comparatively higher resource. For example, both the Alemannic version of Wikipedia and the 'GSW' tagged data in the crawled OSCAR corpus (Ortiz Suárez et al. 2019) are almost exclusively in Swiss German, not Alsatian. The second challenge of Alsatian is its **dialectal variation**. First of all, Alsatian is not a single language but a continuum of languages, with several dialectal areas (Figure 9.1), each with different linguistic traits and vocabulary. Then, another difficulty

[4] Non parsable lexicons and phrasebooks at `http://www.olcalsace.org`.

is introduced by its graphic variation, as Alsatian does not have an orthography, though several conventions have been introduced since 2003, such as the GRAPHAL system (Hudlett 2004), then the ORTHAL one (Zeidler and Crévenat-Werner 2008), updated in 2016. This means that a given word might be spelled differently depending on the region and chosen (or not) spelling system. For example, Bernhard and Steiblé (2015) underline four different forms for the word 'elbow', *Elleboje*, *Ällabooga*, *Elleböje* and *Ellaboja*, all valid and in use. Usual methods for managing Alemannic dialectal variation either imply normalizing all vocabulary to a standard form (Honnet et al. 2018; Millour 2020; Samardzic et al. 2015) or, on the other hand, include identifying or generating variants through rule based (Bernhard and Steiblé 2015; Millour 2020; Scherrer and Rambow 2010b) or neural systems (Schmidt et al. 2020).

We study the creation of bilingual lexicons between Alsatian and the majority language of its regional environment, which, as we saw above, is **French** (FR), a Romance langage (both are remotely related, being Indo-European languages). As this pair is low-resourced, we want to study leveraging a related higher-resourced language, and choose **German** (DE), a West Germanic, High German language, with available lexicons to and from French. To help manage dialectal variation, we look for a closely-related language with the same behavior, and settle on its closest relative, **Swiss German** (GSW), which is more studied, as well as undergoing considerable variation too. Lastly, to study the impact of multilinguality, we add another related language, intermediate both in terms of closeness and data size: **Luxembourgish** (LB), a West Germanic, High German, Central Franconian language which underwent French influences in its vocabulary.

9.1.2 Task

Though lexicons have been induced with cognates (Mann and Yarowsky 2001; Scherrer and Sagot 2014) or a variety of other methods (see Irvine and Callison-Burch (2017) for a survey), in this section, we use character-level machine translation methods (both statistical and neural) to generate, from a given monolingual lexicon, its counterpart(s) in a language of interest. We study more specifically the generation of French-Alsatian lexicons. We want to compare the different aspects of multilinguality we saw earlier, such as using a higher resourced language pair for pivoting, leveraging transfer learning in multilingual models of related languages, and comparing all this to the robust and efficient statistical models.

9.1.3 Hypothesis

We suppose that applying our character based machine translation system to the task of parallel lexicon induction might prove interesting for several reasons. First, we showed that using our system in n-best allowed to take into account ambiguity linked to variation, especially in the case of phonetic variation and ambiguity. Meloni et al. (2021) had also shown that using orthography as a proxy for phonetic relations in the case of historical word reconstruction gives as good, if not better, results. In the case of Alsatian, spelling variation can definitely be seen as different realisations of a common original phonetic form (such as in the previous example). In the previous section, we also managed to mitigate the low-resourced aspect of our previous task by leveraging multilinguality, and we expect it could work in this context too, if we find enough multilingual related data. However, there will be the added difficulty of changing language family, as

we want to generate French-Alsatian (Romance-Germanic) lexicons. It will likely be harder to do character-based 'translation' on out-of-context words than it would be on sentences, as NMT on sentences leverages information from implicit learned semantic context as well as orthographic similarity, where here we only know the surface form.

9.2 Gathering data

One of the main difficulties of our experimental setup is the data gathering. For this step, we explored available datasets, either academic or generated by hobbyists.

9.2.1 General process

Once a dataset was found, its word pairs were extracted, cleaned and parsed, to separate parallel lexicons in all available languages. (We removed the determiners, as they were not systematically present for nouns.) Then, all datasets of a given language pair were concatenated and duplicates were removed. This time, there was no need for phonetization, and pairs containing several forms for a given word were kept.

9.2.2 'Non-academic' data sources

Though this section contains what could be called 'hobbyist resources', the sheer size and quality of some of them made me call them 'non-academic' instead (and even that feels unfair, given the level of investment and expertise of some authors).

Alsatext Alsatext[5] is a quadrilingual Alsatian dictionary, with translations in French, German and English. The author was contacted regarding data licensing but never replied. This site's XML was parsed, and items were extracted depending on their item class, keeping only Alsatian singular words with their French and German translations. This allowed to generate DE-FR, ALS-FR and ALS-DE parallel lexicons. The author has been contacted through the contact form of their website to get authorisation.

ElsassischWarterbuach The ElsassischWarterbuach (Dictionary of Alsatian) is a (now defunct) website,[6] compiled by the late André Nisslé. For this website, all pages were successively queried from the WebArchive in HTML format. All lexicon data was then extracted from the 'table' section of the page, with Alsatian words in bold ('b' tags), and French in italic ('i' tags). Words were then split (on dashes and slashes), filtered on length ratio (no word pair was allowed a bigger size difference than 3), and encoding errors were corrected (&u for ü). This allowed to generate parallel ALS-FR lexicons.

[5] http://www.alsatext.eu/dictionnaire.php.

[6] Accessible via the 2016 version of the WebArchive: http://web.archive.org/web/20160403054628/http://culture.alsace.pagesperso-orange.fr/dictionnaire_alsacien.htm.

Wörterbuch Französisch Deutsch The Wörterbuch Französisch Deutsch (German-French Dictionary) is a (also now defunct) website,[7] compiled by the late Jean-Paul Cronimus.[8] For this website, all pages were successively queried from the WebArchive in HTML format. All lexicon data was then extracted from the 'text' section of the page, with German words in bold ('b' tags), and French as non bolded children of the current item tag. Words were then split (on semi-colon), content between parentheses or brackets removed with regexes, and pairs were filtered on length ratio (no word pair was allowed a bigger size difference than 3). This allowed to generate parallel DE-FR lexicons.

Langenscheidt Vokalbeltrainer On the Langenscheidt Vokabeltrainer website, several datasets are available, among which a bilingual French-German lexicon used for the preparation of the Abitur exam.[9] The website owner (Jörg-Michael Grassau) as well as the original lexicon author (Benedikt Buettner) kindly gave their authorisation for academic use by email. All lexical data was extracted as a raw text file, then split on tabulations to separate French and German words. Data in brackets was removed, and words were split on commas. Lastly, if any of the words took more than 50 characters (therefore was more of a sentence), it was removed.

Nafoku The Nafoku website, coded by Sabine Rennwald, contains a bilingual French-German lexicon of 3700 entries.[10] The author was contacted regarding data licensing but never replied. The page text was copied, then word indexes, gender or case indications were removed from the data ('m' for masculine, for example), as well as indications contained between parenthesis. The data was split on the equal sign separating French and German words, which were then split on commas to separate synonyms. All words were then saved, which allowed to generate a bilingual DE-FR lexicon.

Lingvaro Lingvaro is a freely available cross-lingual hobbyist dictionary under CC BY-NC-SA 3.0 licence by Luis Quesada Torres, in Spanish, English, German, Swiss German (likely the Zürich dialect spoken by the author), and Esperanto.[11] Data from the tsv file was separated in sentences (discarded) and words, among which were only kept the unique word pairs available in German to Swiss German. All words were split on slashes to separate synonyms (as in German einige, corresponding to GSW 'epaar / espaar / paar / äinigi / etli'), and incidental word pairs were kept, which allowed to generate a DE-GSW lexicon.

Wörterbuch Berndeutsch The Wörterbuch Berndeutsch is a bilingual Swiss German - German lexicon, focused on the Bern Swiss German dialect, developed by Eduard Muster.[12] The author was contacted regarding data licensing but never replied. Contents were extracted as raw text, split on the equal sign for language, then each word group was split on numbers, semi

[7] Accessible via the 2016 version of the WebArchive: `http://web.archive.org/web/20160611154300fw_/http://cronimus.apinc.org/dico/index.htm`.

[8] Technically, the website includes a licence of sorts, with a text saying that the resource is usable freely (in both senses) by individuals, or pedagogical services belonging to the French Education Nationale or in a contract with the French State, though distribution is forbidden without a previous permission.

[9] `http://www.vokabeln.de/v3/vorschau/Franzoesisch_Abitur.htm`.

[10] `http://nafoku.de/wbuch/f-basis.htm`.

[11] `http://github.com/lquesada/Lingvaro`.

[12] `http://www.edimuster.ch/baernduetsch/woerterbuechli.htm`.

colons or commas to separate synonyms; pairs with a bigger than 1:3 length ratio were removed. This allowed us to generate a GSW-DE lexicon.

9.2.3 Institutional dataset

Lëtzebuerger Online Dictionnaire The Lëtzebuerger Online Dictionnaire (Online Luxemburgish Dictionary) is freely available online[13] under a CC0 licence. It was developed by the Luxembourg ministry of culture, and is provided as XML. Items were split using 'lod:ITEM' tags, then Luxembourgish words were extracted using the 'lod:ITEM-ADRESSE' tags and the French and German words using the 'lod:EQUIV-TRAD-FR' and 'lod:EQUIV-TRAD-ALL' tags respectively. Then data in parenthesis was removed, and word pairs with a bigger than 1:3 length ratio were removed. This allowed us to generate three bilingual lexicons: DE-FR, FR-LB and DE-LB.

9.2.4 Academic datasets

Wörterbuch der elsässischen Mundarten The Trier Center for Digital Humanities kindly supplied us with the OCRised version of the 1900 alsatian dialects dictionary 'Wörterbuch der elsässischen Mundarten'.[14] As first, the format was sadly non standard (an in-house variation of TEI), second, it contained OCR errors and third, it was too different in nature from the rest of our data (dictionary rather than lexicon), we were unable to parse it and use it, though it could be used for further work on dialectal variation.

Multilingual lexicon linked to BabelNet synsets Delphine Bernhard was kind enough to provide the trilingual word list of cognates she compiled (German, French, Alsatian) and referenced in Bernhard (2014).[15] For each word group of each language, information in brackets was removed, then word groups were split on semicolon to separate synonyms. This allowed us to generate three datasets (DE-FR, ALS-DE, ALS-FR).

Annotated Corpus for the Alsatian Dialects This Alsatian corpus (CC BY-SA 4 licence) from the RESTAURE project (Bernhard et al. 2018a) is annotated in POS, and contains lemma and French gloss for each Alsatian word.[16] We parsed all files present, keeping for each the uninflected lemma and French gloss (after discarding punctuation marks lemmas), and generated a bilingual ALS-FR dataset.

Diverse sources The last corpus found was a zipped file available for download on the Bisame website project (for Alsatian myriadisation, no longer maintained), by Alice Millour.[17] It

[13] `http://data.public.lu/fr/datasets/letzebuerger-online-dictionnaire-raw-data/`.
[14] API: `http://woerterbuchnetz.de/?sigle=ElsWB`, more information about the resource: `http://fr.wikipedia.org/wiki/Dictionnaire_des_parlers_alsaciens`.
[15] `http://nakala.fr/10.34847/nkl.3f9b2i11`.
[16] Raw files at `http://zenodo.org/record/2536041`.
[17] `http://bisame.paris-sorbonne.fr/recettes/downloads`.

is likely this dataset was created by Delphine Bernhard, and it was constituted of several texts annotated in BROWN format, UD, and provided as raw text. We extracted lemmas in Alsatian and their French gloss by taking the correct columns in the file, removing punctuation and content in parenthesis as well as unknown words, and splitting on either slashes (lemmas) or semicolons and hashes (inflected lemmas). This allowed us to extract a bilingual ALS-FR dataset.

9.3 Experimental setup

9.3.1 Final data statistics

From ALS to	ALS	DE	FR	GSW	LB
#words	48153	11468	73675	-	-
#characters	476747	174103	1369140	-	-
#unique characters	82	77	88	-	-
Avg word length	10.21	7.60	9.66	-	-
From DE to	ALS	DE	FR	GSW	LB
#words	11468	91461	100024	4990	22624
#characters	174103	1080393	2406520	74861	431237
#unique characters	77	71	82	62	72
Avg word length	7.60	11.89	12.58	7.61	9.57
From FR to	ALS	DE	FR	GSW	LB
#words	73675	100024	92123	-	22652
#characters	1369140	2406520	1225167	-	430319
#unique characters	88	82	84	-	75
Avg word length	9.66	12.58	14.56	-	9.75
From GSW to	ALS	DE	FR	GSW	LB
#words	-	4990	-	4370	-
#characters	-	74861	-	31530	-
#unique characters	-	62	-	60	-
Avg word length	-	7.61	-	7.29	-
From LB to	ALS	DE	FR	GSW	LB
#words	-	22624	22652	-	22322
#characters	-	431237	430319	-	211532
#unique characters	-	72	75	-	71
Avg word length	-	9.57	9.75	-	9.50

Table 9.1: Dataset statistics.

Our final concatenated dataset varies considerably in size depending on language pairs (Table 9.1): our biggest ones are ALS-FR (73K word pairs) and DE-FR (100K word pairs), then DE-LB and FR-LB at 20K, then finally ALS-DE at 10K and DE-GSW at 5K. We split our data in train/dev/test sets (85%/7.5%/7.5%). We pick monolingual data by extracting the set of all words available in a given language.

9.3.2 Models of interest

Preliminary hyper-parameter search We first try to find the best possible hyper-parameters for our language combinations. We compare recurrent and Transformer-based[18] models, both in a bilingual setup for all languages pairs (B-NMT) and in a massively multilingual setup (M-NMT) trained on all language pairs available (for 20 epochs with an Adam optimizer). However, as our datasets are several orders of magnitude bigger than they were for cognate prediction, so are our computation times. Doing a full hyperparameter search is therefore not possible for all our models, seeds, and language pairs - we select specific cases depending on their computation time. We manage to generate a full bilingual hyperparameter search for our smaller ALS-DE, DE-GSW, GSW-DE and LB-FR, as well as a comparison of hidden size-embedding size for the bigger DE-FR and FR-DE sets. For multilingual models, which are considerably bigger as they include all available data, we only run embedding size vs hidden layer size experiments. After comparing the bilingual and multilingual parameters trends, we conclude that (1) Transformers are still underperforming and (2) decide to use bilingual models results on the hyper-parameter search as proxy for the multilingual models, as they seem to correlate well for the available results.

SMT	B-NMT	M-NMT
-	$L_1 \rightarrow L_2$.	$L_1, L_2 \rightarrow L_1, L_2$.

Table 9.2: Model type reminder (see Section 5.1).

Chosen models We therefore train all models with one encoder layer, one decoder layer, an embedding dimension of 28 and hidden size of 90, a batch size of 30 and learning rate of 0.001, and varying language combinations: we use either French and all Germanic languages (**M-NMT**$_5$: ALS-DE-GSW-LB and FR), French and Upper German languages (**M-NMT**$_4$: ALS-DE-GSW and FR), French, Alsatian, and the highest-resourced other related language, German (**M-NMT**$_3$), and bilingual models (**B-NMT**). Then, we compare our raw results to zero-shot transfer experiments, and using FR-DE to zero-shot FR-ALS and DE-FR to zero-shot ALS-FR.

Other preliminary experiments We also compare using our best performing setup with added monolingual data, which did not increase performance, and with a shared encoder and a shared decoder across all Germanic languages in our multilingual model, which actually decreased performance two-fold;[19] we therefore only focus on our simplest and best performing setup.

[18] Since we have more data, maybe Transformer-based models will perform better.

[19] It is likely the performance decrease is actually due to information diffusion across our languages of interest.

9.4 Results

9.4.1 Raw results

In our results, we display the language pairs common to all setups: ALS-FR, ALS-DE and DE-FR (Figure 9.2). Results on ALS-DE allow us to study how well the model performs on two close languages from the same family, with an average amount of data (20K pairs). Performance on DE-FR allows us to see how well the model performs on a more distant pair with a high amount of data (100K pairs) but no ambiguity, where performance on ALS-FR adds this extra ambiguity into account while containing a fair amount of data (70K pairs). From our previous experiments on cognates, we could expect our models to perform best on ALS-DE due to linguistic proximity, then on DE-FR, an unambiguous situation, then lastly on ALS-FR, an ambiguous situation (though n-best performance could mitigate this last difference).

Comparing language directions We first observe that our BLEU scores on ALS-DE are similar to those obtained for our related Romance languages (between 62 and 67.2 to ALS and 68.7 and 78 BLEU to DE, in 1-best, depending on the setup). It seems harder to predict to ALS than from it, which is not surprising, as the situation presents an asymetrical ambiguity,[20] where a given letter in German can correspond to several in Alsatian depending on dialectal spelling, where the opposite is not true sine German spelling is regular; this difference in ambiguity is reduced but not entirely erased by predicting in n-best, with 10-best results still lower to ALS than from it. We can therefore first say that our model has been capable of learning to translate character-level lexicons, and that our models for predicting cognates transfer quite well to ambiguous situations such as predicting a language with high dialectal variation, even with a relatively small dataset (10K).

However, one of our main goals was to use our models in a cross-family setup, predicting ALS from FR and vice-versa. The results here are considerably lower: DE-FR barely reach 30 BLEU, and ALS-FR only surpasses 10 BLEU using statistical methods – our character level translation models do not seem well adapted to cross-family setups. The difference between using FR with ALS or with DE can likely be explained by a combination of data size difference (30%) and ambiguity (ALS containing variation and DE not).

Comparing models types The respective performance of our models varies depending on the language pair: on ALS-DE (our closest pair), the worst performing models are our statistical baselines, and our best performing models are the M-NMT$_3$ or M-NMT$_5$ models, with a difference of about 5 to 10 BLEU between worst and best (which represents a score variation between 7 and 12%). It is interesting that for ALS-DE, the best overall setups are either M-NMT$_3$, which include pairs with FR, therefore adding a maximum amount of ALS data in a single step, or M-NMT$_5$, containing the maximum number of Germanic languages overall.

[20] This ambiguity is much like predicting from and to a parent language in our previous experiments

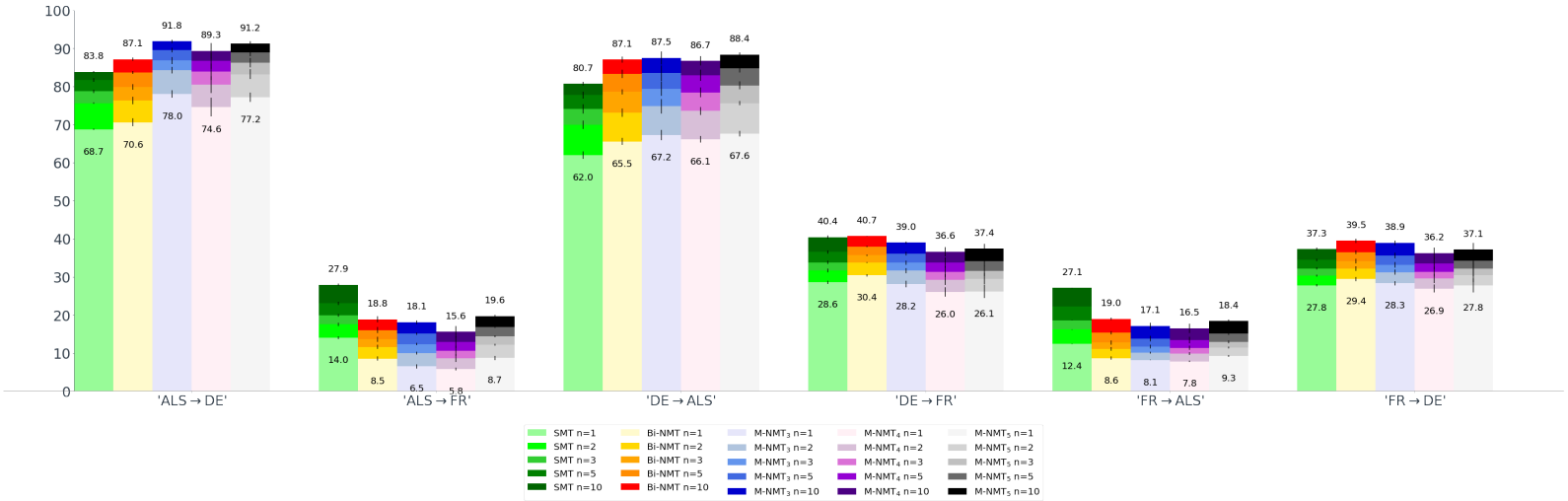


Figure 9.2: BLEU scores comparison for Germanic-French lexicon induction. Colours indicate the model type for a language pair: SMT in green (col 1), Bi-NMT in orange (col 2), M-NMT$_3$ in blue (col 3), M-NMT$_4$ in purple (col 4), M-NMT$_5$ in grey (col 5). Colour shades indicate the value of n in n-best predictions (1, 2, 3, 5 and 10 from bottom to top).

For DE-FR, all models behave quite similarly, with a maximum score difference of about 4 BLEU: the best performing model is the Bi-NMT model, with SMT and M-NMT$_3$ close behind (and M-NMT$_5$ when going to DE), at a 2 BLEU points difference (7%). Across families, and with enough data, the best performance goes to bilingual neural models, specialised on the precise task.

For ALS-FR, however, the best model by far (between 50 and 60% increase in performance, or 4 to 6.5 BLEU), is the statistical baseline, followed by our Bi-NMT and M-NMT$_5$ models. As the performance is worst to FR than to ALS, though the latter is the more ambiguous case, it could be that ambiguity does not actually play that much a role, and that the bad performance in both directions of the ALS-FR prediction is actually due to data size: the ALS-FR pair could be just below the necessary data threshold (for lexicons) for neural methods performance, which DE-FR has reached (being 30% bigger).

Interestingly, multilinguality seems to either penalise the "out of family" language (French) when too little data is added (M-NMT$_3$, M-NMT$_4$), or help it with enough new languages and data (M-NMT$_5$), much like we saw when going to/from Romanian and Aromanian in our previous experiments.

Analysing predictions To understand the variation between language pairs in more depth, as well as better analyse our models' capabilities, we look at neural predictions for our different language pairs. We observe that our models have learnt correspondences between source and target languages, as well as a target word language modelling aspect, predicting plausible words. For example, for our best performing Bi-NMT model, German ***resolut***, which corresponds to French ***résolu*** was predicted in 10-best as résolute, résoluteur, résolutif, résolut, résolutive, résolution, résolutile, résoluté, résolume or résout, strictly wrong, but phonotactically correct (and sometimes semantically close) French words, whereas German ***Muschel***, translatable in French as ***conque*** (our target) or ***moule*** was predicted as muschèle, muschel, musille, muschele, muschule, Muschel, mousse, moule, muchel or musure, with the correct word (moule) appearing along phonotactically plausible words, and German inspired words. Interestingly, the different incorrect predictions seem to reflect a continuum of phonetic correspondences, from German sounding muschel/muchel, to muschèle/muschele, to muschule, to French sounding musille, for example. This could reflect that the model learned phonetic patterns from different steps in the language history, since we provided it with a bilingual lexicon containing, by essence, normal words as well as both cognates (having lived the full phonological history of their languages) and borrowings (only carrying phonetic evolution and correspondances having occured after their arrival).

However, when the word shapes are too different while going from a Germanic language to French, the neural models get lost and tend to produce previously seen words. For example, when going from German ***abbüßen*** to French ***expier***, the model gets stuck on the fact that initial German 'ab' can be linked to French prefix 'en' or 'dé', and uses it to start its predictions: entraîner, conserver, départer, changer, déchanger, entrairer, départir, déchancer, entraire, and entrer, some of which exist, but none of which is correct. It is likely that cross-family language translation would need a much bigger amount of data to avoid pseudo-overfitting situations such as this one. We can therefore say that, for character-level lexicon translation, our neural models learn an aspect of phonotactic language modelling, but that 100K word pairs are not enough to do cross-family prediction.

9.4.2 Zero-shot transfer results

To study how specific our neural encoders and decoders are, and how easy it is or not to leverage the 'better' performing FR-DE encoder/decoder pairs for ALS, we try some zero-shot experiments.

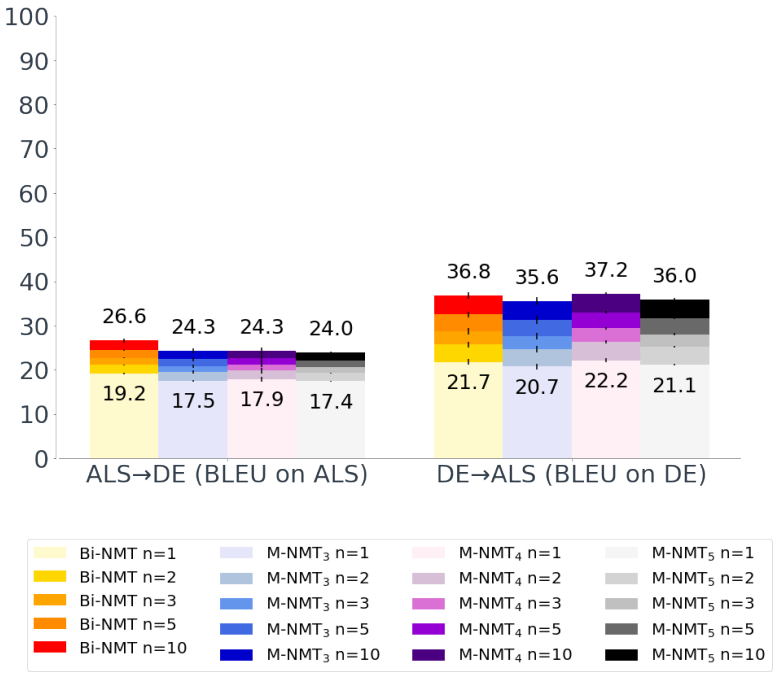


Figure 9.3: BLEU scores for zero shot identity experiments. Colours indicate the model type for a language pair: Bi-NMT in orange (col 1), M-NMT$_3$ in blue (col 2), M-NMT$_4$ in purple (col 3), M-NMT$_5$ in grey (col 4). Colour shades indicate the value of n in n-best predictions (1, 2, 3, 5 and 10 from bottom to top).

We first want to see how close to an ALS identity function DE is, or, in other terms, if DE can be used as a substitute for ALS. We therefore look at the BLEU between the DE prediction of our trained ALS-DE model and the source ALS word (is German good pseudo-Alsatian?), as well as between the ALS prediction of our trained DE-ALS model and the actual DE source (is Alsatian a good pseudo-German?). Going both ways will allow us to see the difference between predicting DE from an ALS decoder, used to variation, and ALS from a DE decoder, better performing.

We observe (Figure 9.3) that results are overall very low (for what is an identity prediction), with a BLEU around 17-20 points in 1-best, which reflects the strong differences between ALS and DE. This score goes up to 26 when predicting using the DE decoder to predict ALS, and 36 when predicting DE from the ALS decoder in 10-best. The ALS decoder, more used to variation, produces a wider range of answers than the DE decoder, which cannot anticipate the expected variation for ALS. To summarize, it is unlikely that using DE as a proxy for ALS in the hope of capitalizing on its 'better' performance to FR in zero shot would work. We try it nonetheless.

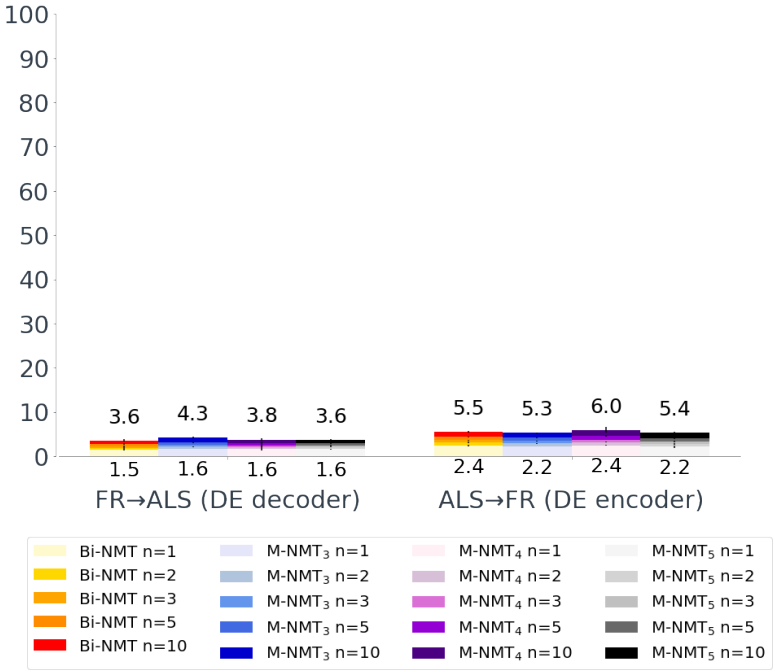


Figure 9.4: BLEU scores for zero shot experiments. Colours indicate the model type for a language pair: Bi-NMT in orange (col 1), M-NMT$_3$ in blue (col 2), M-NMT$_4$ in purple (col 3), M-NMT$_5$ in grey (col 4). Colour shades indicate the value of n in n-best predictions (1, 2, 3, 5 and 10 from bottom to top).

On Figure 9.4, we observe that the BLEU score obtained when using the DE encoder (resp. decoder) as a substitute to predict ALS⟶FR (resp. FR⟶ALS) are even lower than just focusing on ALS-FR directly. This confirms that German is not a good proxy for Alsatian in zero-shot, as such. Going from a language family to another, as well as using a high-resourced but still quite different reference language for zero-shot, and the inherent ambiguity variation of Alsatian combine together to prevent this task from properly working.

9.5 Managing dialectal variation

Contrary to our initial expectations, it appears very difficult to directly produce a French-Alsatian lexicon from a char-based MT model trained on Germanic languages and French, as the performance is just too low. However, when going from DE to ALS, we observe that the model learns spelling variations motifs, such as when going from DE *Entscheidungsträger* to FR *Äntschaidungstrager*, where the model predicts Entschaidungschträger, Äntschaidungsträjer, Entschaidungsträjer, Entschaidungschträga, Äntschaidungschträga, among shorter variations, with the initial 'ent' appearing both as 'ent' and 'änt' and the ending 'ager' as 'äger', 'äjer' or 'äga' (correspondences which exist, as mentioned in Section 9.1.1, with 'elbow' being written *Elleboje*, *Ällabooga*, *Elleböje* and *Ellaboja*.).

9.6 Conclusion

After compiling and extracting bilingual lexicons in our languages of interest (ALS, DE, FR, GSW, LB), we experimented with different setups to find the best parameter combination for cross-family character-level MT models. We found that the best architectures were the simplest (without added monolingual data, and without sharing components), either statistical or neural recurrent (Transformers still under performing). When studying models and language pairs more specifically, we observed that intra-family prediction on Germanic languages performed as well as our previous experiments on Romance families, but that cross-family prediction did not perform well, plausibly because of a lack of data which caused overfitting. We also observed that our models captured what seems to reflect phonotactic comprehension, and were able to mitigate ambiguity for Alsatian when coming from German. We therefore suppose that, using an already existing bilingual French-German lexicon, it could be possible to induce plausible Alsatian variants from the German words, using a character-based machine translation approach trained on enough data, and then to check predictions by looking for occurences in monolingual Alsatian corpora, which would circle back to previous work (Mann and Yarowsky 2001). However, though Germanic character based multilingual machine translation models seem to help the prediction of Alsatian, further work in this direction is needed.

CONCLUSION AND OUTLOOK

> Man is poorly designed for collating masses of data, but is able to bring a wide range of knowledge and interpretation to a single item of data at any one time. The machine, on the other hand, betrays a certain inanity when it comes to matters of interpretation, since it has no knowledge of the world, and is thereby obliged to form its interpretations on mere surface appearances, [while its capacity] to collate and search, on the other hand, may well arouse our admiration. As for the combination of man and machine, we are only just beginning to realize the potentialities that this combination has for the world of scholarship, and especially for the discipline of Linguistics.
>
> Hewson (1973)

The primary purpose of the current dissertation was first, to study if and under which conditions neural networks could be used for the task of cognate prediction, then determine more precisely what they learnt on this task. In historical linguistics, cognates are words that descend in direct line from a common ancestor, called their proto-form, and therefore are representative of their respective languages' evolutions through time, as well as of the relations between these languages synchronically. As they reflect the phonetic history of the languages they belong to, they allow linguists to better determine all manners of synchronic and diachronic linguistic relations (etymology, phylogeny, sound correspondences) (Durkin 2015). The formalisation of the empirical sound change regularity rule by the Neogrammarians (Osthoff and Brugmann 1878) represented an important step in the formal study of cognates.

As sounds change regularly through their languages' history, cognates of given languages tend to be linked, and identified, through systematic correspondence patterns, and neural networks, being especially good at learning latent patterns, seemed to be a good match for this task. At the beginning of this work, in 2019, they had started being used for cognate identification (Frunza and Inkpen 2006; Frunza 2006; Kumar et al. 2017; Sepúlveda Torres and Aluísio 2011) and cognate prediction (Beinborn et al. 2013; Dekker 2018; Mulloni 2007; Nguyen et al. 2018; Wu and Yarowsky 2018), with two caveats. First, the word *cognate* can reflect several concepts depending on the field (see Section 1.2.2), and cognate identification and prediction papers from the machine translation field or language learning studies do not usually concern themselves with historical linguistics constraints, therefore not trying to capture the specific patterns we are interested in. Secondly, prior work specifically on historical word prediction seemed to indicate that there were inherent limitations (data size, complexity, ambiguity) which could prevent it from being learnable by neural networks, especially those we were interested in (Dekker 2018) — though one paper obtained promising results (Meloni et al. 2021).

In the current dissertation, we sought to methodically study the applicability of machine

translation inspired neural networks to historical word prediction, relying on the surface similarity of both tasks, modelling sequence to sequence relations. In order to differentiate between the hypothesis which could justify prior failure (theoretical differences too big, lack of model optimisation, too small data size, task inherently too complex), we first created an artificial dataset inspired by the phonetic and phonotactic rules of Romance languages, which allowed us to vary task complexity and data size in a controlled environment (Fourrier 2020; Fourrier and Sagot 2020a). This led to the conclusion that, first, neural machine translation inspired neural networks could theoretically be used for the task, as long as they were optimised and the data size was above a threshold, and second, that historical word reconstruction was not a symmetrical task, with parent to child being the easiest direction to predict, child to parent the most ambiguous, and child to child the most theoretically similar to machine translation.

We then extended our work to real datasets, and updated an etymological database to gather a correct amount of data (Fourrier and Sagot 2020b), then studied first the transferability of our conclusions on model success and optimisation to real data, then the applicability of a number of data augmentation techniques to the task, to try to mitigate low-resource situations. We concluded that above a certain data size, and after optimisation, the best possible setup was using multilingual recurrent neural networks. It performed better than using other models (Transformer-based) or other techniques, such as pretraining and back-translation, also leveraging added data (Fourrier et al. 2021).

We finally investigated in more detail the advantages of using multilingual neural networks compared to other setups, by looking at six possible instanciations of our neural models, bilingual or multilingual, with or without monolingual data, sharing components or not, using statistical methods as a baseline. We first confirmed that, on the surface, they seem to capture language relatedness information and phonetic similarity, confirming the work of Meloni et al. (2021). We then discovered, by probing them, that the information they store is actually more complex: our multilingual models actually encode a phonetic language model, and learn enough latent historical information to allow the reconstruction of the (unseen) proto-form of the studied languages as well or better than bilingual models trained specifically on the task (Fourrier and Sagot 2022). This latent information is likely the explanation for the success of multilingual methods in the previous works. Now...

WHAT comes next?

Extension to other languages During this work, we studied and analysed our models on one single language family, the Romance language, which is already extremely well studied, and not the one having undergone the most complex developments. Further work will therefore need to study other language families, in order to see if the results obtained in this manuscript can be generalized. Preliminary work on the training datasets of the SIGTYP 2022 cognate prediction competition (covering 10 varied language families, with datasets containing between 4 and 19 languages)[1] showed that our models, in the 'best' possible settings (between 107 word pairs for 9 languages and 944 word pairs for 16 languages, still very low-resourced situations compared to the data studied in this dissertation) barely reached a performance level equivalent to a linguistically motivated baseline, without a clear difference between bilingual models and multilingual ones. Data size still remains a hard limit for our models. Next steps to adress this issue would therefore be working either on data collection and training for other languages fam-

[1] github.com/sigtyp/ST2022.

ilies, or investigating other aspects of data augmentation, and likely designing new techniques fitted to this task (such as pretraining models with artificial data reflecting the statistics of the studied languages).

Cognate detection Cognate prediction models could then also be used as plausible cognate generators, as in Bodt et al. (2018) and Bodt and List (2019), in which case plausible generated cognates could either be investigated in the field, or looked for in digitized manuscripts (where the natural ambiguity management of our models could also allow to mitigate natural variation), looping back to cognate identification.

Proto-form reconstruction If results were to be convincing enough on other families, it would allow us to try to replicate model probing, and investigate the latent representation for other language families using similar probes to the previous chapters. Another direct extension could also be to find a way to directly extract the latent information we uncovered through probing tasks from our neural networks. Preliminary unpublished experiments trying to differentiate intermediate representation of phones depending on their contexts of origin have not been conclusive so far, but further work in this direction is obviously needed.

Borrowing detection The models we developed could also be used to study the differences between cognate and borrowings evolution, contributing to borrowing detection (Ciobanu and Dinu 2015; Ciobanu and Dinu 2019; List and Forkel 2022). For example, an application could be the study of the different latent sound correspondences learnt by a model trained on cognate prediction versus one trained on borrowing prediction.

Of course, in the different extensions underlined, neural networks would only be used as allies to linguists, who would make the final judgement on the quality of their predictions and reconstructions.

Other historical linguistic tasks Cognates and borrowings can also be useful outside of their prediction and identification ***stricto sensu***. Several other historical linguistic tasks rely on historical words, such as sound changes rules validation (Hewson 1973; Smith 1969) or phylogenetic recontruction, from reconstructing language trees (Dekker 2018; Greenhill 2011; Jäger and List 2016; Rama et al. 2018) to understanding the genetic relations of isolate languages (Hantgan and List 2018). Trained cognate prediction models, which we demonstrated are good at handling ambiguity, could provide artificial but plausible data to check hypotheses for these tasks, or, when validated against monolingual lexicons, provide new plausible cognate sets to help for such tasks.

Semantic studies In the theoretical background of this manuscript, we saw that the regularity of sound changes (the mechanisms behind cognates and borrowings identification) are also at the root of semantic change studies (Durkin 2015). Semantic change studies studies meaning variations between languages or dialects, which has been done using word embeddings as proxies for word meaning, both diachronically on real or artificial data (Martinc et al.

2020; Montariol and Allauzen 2021) and synchronically at the sentence or word level (Beinborn and Choenni 2020; Hovy and Purschke 2018). Cognates have been used to study semantic change (Uban et al. 2021), but as cognates and borrowings reflect different aspects of their languages phonetic history, do they differ in their semantic evolution trajectories? If they did, they would constitute, by their very nature, a promising proxy for evaluating semantic change methods. In preliminary work (Fourrier and Montariol 2022), we tried using cognates and borrowings to see if they could also be used for diachronic semantic studies, possibly as markers to study rates of semantic change between languages, though our experiments were too preliminary to show anything more than light trends — it would be interesting to reproduce these experiments, but paying close attention to the respective data sizes of cognates and borrowings between languages, in order to get statistically comparable results. More generally, cognate prediction is the door to a range of diachronic studies outside of phonetics, from morphological evolution to semantics. Optimising cognate prediction models, this time on segmentation (which we did not study), for example by using sub-word instead of character level tokenization, could provide interesting insights on low-resourced cognates morphology, especially if the observed multilingual latent learning also occurs when studying cognates from a morphological standpoint.

Cognates beyond historical linguistics Cognates can also be useful synchronically, for other tasks, and have for example been used in low-resourced machine translation setups, to induce lexicons (Hauer et al. 2017; Mann and Yarowsky 2001), as seeds to align multilingual word embeddings, or to help script decipherement (Luo et al. 2021).

Beyond cognates In the opening Chapter, we went past cognates as such, and focused on lexicon induction, using our models to try to get the best cross-family results on the task, while studying the impact of available languages on prediction. We observed that character-level machine translation performs considerably better inside a given language family (reaching the performance we had in previous experiments on Romance cognates), even mitigating natural dialectal variation. Experiments on dialectal variation management could be extended, by producing plausible word forms from a neighbouring language, then training a classifier to determine the plausible geographic origin of the given predictions. To predict words between unrelated languages (since using our models cross-family gave poor results) we suppose that similar multilingual architectures could be trained, language family per language family, then connected through "bridge-models" across intermediate representations. Preliminary experiments seem to show that this would be an interesting direction for multilinguality management, especially in low-ressource situations.

BIBLIOGRAPHY

[1] I. Abdulmumin, B. S. Galadanci, and A. Isa. "Enhanced back-translation for low resource neural machine translation using self-training". In: *International Conference on Information and Communication Technology and Applications (ICTA 2020)*. Dec. 2020. DOI: 10.1007/978-3-030-69143-1_28 (cit. on p. 52).

[2] D. Adams. *So Long, and Thanks for All the Fish*. 1984 (cit. on p. 243).

[3] R. Aharoni, M. Johnson, and O. Firat. "Massively Multilingual Neural Machine Translation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 3874–3884. DOI: 10.18653/v1/N19-1388 (cit. on p. 52).

[4] D. Alvarez-Melis and T. Jaakkola. "Gromov-Wasserstein Alignment of Word Embedding Spaces". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 1881–1890. DOI: 10.18653/v1/D18-1214 (cit. on p. 54).

[5] C. Anderson, P. Heggarty, and The CoBL Consortium. *Cognacy in Basic Lexicon database*. 2020 (cit. on p. 17).

[6] C. Anderson, T. Tresoldi, T. Chacon, A.-M. Fehn, M. Walworth, R. Forkel, and J.-M. List. "A cross-linguistic database of phonetic transcription systems". In: *Yearbook of the Poznan Linguistic Meeting* 4.1 (2018), pp. 21–53. DOI: 10.2478/yplm-2018-0002 (cit. on p. 11).

[7] M. Artetxe, G. Labaka, and E. Agirre. "Learning bilingual word embeddings with (almost) no bilingual data". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 451–462. DOI: 10.18653/v1/P17-1042 (cit. on pp. 52, 129).

[8] D. Ataman, O. Firat, M. A. Di Gangi, M. Federico, and A. Birch. "On the Importance of Word Boundaries in Character-level Neural Machine Translation". In: *Proceedings of the 3rd Workshop on Neural Generation and Translation*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 187–193. DOI: 10.18653/v1/D19-5619 (cit. on p. 53).

[9] M. Aulamo, S. Virpioja, and J. Tiedemann. "OpusFilter: A Configurable Parallel Corpus Filtering Toolbox". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, July 2020, pp. 150–156. DOI: 10.18653/v1/2020.acl-demos.20 (cit. on p. 39).

[10] S. Auroux, E. F. K. Koerner, H.-J. Niederehe, and K. Versteegh. *History of the Language Sciences/Geschichte der Sprachwissenschaften/Histoire des sciences du langage*. Walter de Gruyter, 2008 (cit. on p. 3).

[11] B. Babych. "Graphonological Levenshtein Edit Distance: Application for Automated Cognate Identification". In: *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*. 2016, pp. 115–128 (cit. on pp. 15, 16).

[12] D. Bahdanau, K. Cho, and Y. Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *Proceedings of the 3rd International Conference on Learning Representations*. Ed. by Y. Bengio and Y. LeCun. 2015 (cit. on pp. 29, 31, 34, 50, 61, 87, 89).

[13] F. Bane and A. Zaretskaya. "Selecting the best data filtering method for NMT training". In: *Proceedings of Machine Translation Summit XVIII: Users and Providers Track*. Virtual: Association for Machine Translation in the Americas, Aug. 2021, pp. 89–97 (cit. on p. 39).

[14] S. Banerjee and A. Lavie. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 65–72 (cit. on p. 51).

[15] L. Barbosa and J. Feng. "Robust Sentiment Detection on Twitter from Biased and Noisy Data". In: *Coling 2010: Posters*. Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 36–44 (cit. on p. 37).

[16] M. Baroni, G. Dinu, and G. Kruszewski. "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 238–247. DOI: 10.3115/v1/P14-1023 (cit. on p. 42).

[17] K. Batsuren, G. Bella, and F. Giunchiglia. *CogNet: A Large-Scale Cognate Database*. 2019 (cit. on p. 17).

[18] K. Batsuren, G. Bella, and F. Giunchiglia. "A large and evolving cognate database". In: *Language Resources and Evaluation* (May 2021). DOI: 10.1007/s10579-021-09544-6 (cit. on p. 17).

[19] A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, F. Dalvi, and J. Glass. "Identifying and Controlling Important Neurons in Neural Machine Translation". In: *arXiv:1811.01157 [cs]* (Nov. 2018) (cit. on p. 46).

[20] L. Beinborn and R. Choenni. "Semantic Drift in Multilingual Representations". In: *Computational Linguistics* 46.3 (Nov. 2020), pp. 571–603. DOI: 10.1162/coli_a_00382 (cit. on p. 148).

[21] L. Beinborn, T. Zesch, and I. Gurevych. "Cognate Production using Character-based Machine Translation". In: *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. Nagoya, Japan: Asian Federation of Natural Language Processing, Oct. 2013, pp. 883–891 (cit. on pp. 15, 17, 18, 145).

[22] Y. Belinkov, S. Gehrmann, and E. Pavlick. "Interpretability and Analysis in Neural NLP". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Online: Association for Computational Linguistics, July 2020, pp. 1–5. DOI: 10.18653/v1/2020.acl-tutorials.1 (cit. on p. 45).

[23] E. M. Bender and B. Friedman. "Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science". In: *Transactions of the Association for Computational Linguistics* 6 (2018), pp. 587–604. DOI: 10.1162/tacl_a_00041 (cit. on p. 37).

[24] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜". In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 610–623. DOI: 10.1145/3442188.3445922 (cit. on pp. 41, 44).

[25] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. "A neural probabilistic language model". In: *Journal of machine learning research* 3.02 (2003), pp. 1137–1155 (cit. on p. 41).

[26] D. Bernhard. "Adding dialectal lexicalisations to linked open data resources: the example of alsatian". In: *Proceedings of the Workshop on Collaboration and Computing for Under Resourced Languages in the Linked Open Data Era (CCURL 2014)*. 2014, pp. 23–29 (cit. on pp. 130, 134).

[27] D. Bernhard, A.-L. Ligozat, F. Martin, M. Bras, P. Magistry, M. Vergez-Couret, L. Steiblé, P. Erhart, N. Hathout, D. Huck, C. Rey, P. Reynés, S. Rosset, J. Sibille, and T. Lavergne. "Corpora with Part-of-Speech Annotations for Three Regional Languages of France: Alsatian, Occitan and Picard". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018 (cit. on pp. 130, 134).

[28] D. Bernhard, P. Magistry, A.-L. Ligozat, and S. Rosset. "Resources and Methods for the Automatic Recognition of Place Names in Alsatian". In: *Corpus-Based Research in the Humanities*. Ed. by A. U. Frank, C. Ivanovic, F. Mambrini, M. Passarotti, and C. Sporleder. Vol. 1. Proceedings of the Second Workshop on Corpus-Based Research in the Humanities CRH-2. Vienna, Austria, Jan. 2018, pp. 35–44 (cit. on p. 130).

[29] D. Bernhard and L. Steiblé. "Quand l'oral se fait entendre à l'écrit: alignement de lexiques en l'absence de normalisation graphique". In: *TALaRE 2015-Traitement Automatique des Langues Régionales de France et d'Europe*. 2015 (cit. on pp. 130, 131).

[30] N. Bertoldi and M. Federico. "Domain Adaptation for Statistical Machine Translation with Monolingual Resources". In: *Proceedings of the Fourth Workshop on Statistical Machine Translation*. Athens, Greece: Association for Computational Linguistics, Mar. 2009, pp. 182–189 (cit. on p. 52).

[31] N. Bertoldi, B. Haddow, and J.-B. Fouet. "Improved Minimum Error Rate Training in Moses". In: *The Prague Bulletin of Mathematical Linguistics* 91 (2009), p. 7 (cit. on p. 61).

[32] R. M. Blench. "Archaeology and Language: methods and issues". In: *A Companion To Archaeology.* (2004), pp. 52–74 (cit. on p. 3).

[33] M. Bloodgood and B. Strauss. "Using Global Constraints and Reranking to Improve Cognates Detection". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1983–1992. DOI: 10.18653/v1/P17-1181 (cit. on p. 16).

[34] T. A. Bodt, N. W. Hill, and J.-M. List. "Prediction experiment for missing words in Kho-Bwa language data". In: *Open Science Framework Preregistration* (2018) (cit. on pp. 4, 16, 17, 82, 129, 147).

[35] T. A. Bodt and J.-M. List. "Testing the predictive strength of the comparative method: an ongoing experiment on unattested words in Western Kho-Bwa languages". In: *Papers in Historical Phonology* 4 (2019), pp. 22–44. DOI: 10.2218/pihph.4.2019.3037 (cit. on pp. 4, 16, 17, 129, 147).

[36] O. Bojar and A. Tamchyna. "Improving Translation Model by Monolingual Data". In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, July 2011, pp. 330–336 (cit. on p. 52).

[37] M. Bollmann. "A Large-Scale Comparison of Historical Text Normalization Systems". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 3885–3898. DOI: 10.18653/v1/N19-1389 (cit. on p. 51).

[38] A. Bouchard, P. Liang, T. Griffiths, and D. Klein. "A Probabilistic Approach to Diachronic Phonology". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 887–896 (cit. on p. 16).

[39] A. Bouchard-Côté, T. L. Griffiths, and D. Klein. "Improved Reconstruction of Protolanguage Word Forms". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 65–73 (cit. on p. 16).

[40] A. Bouchard-Côté, D. Hall, T. L. Griffiths, and D. Klein. "Automated reconstruction of ancient languages using probabilistic models of sound change". In: *Proceedings of the National Academy of Sciences* 110.11 (Mar. 2013), pp. 4224–4229. DOI: 10.1073/pnas.1204678110 (cit. on p. 17).

[41] P. Boyd-Bowman. *From Latin to Romance in Sound Charts*. Georgetown University Press, 1980 (cit. on pp. 101, 115, 118, 119, 242).

[42] R. Bradbury. *Fahrenheit 451*. 1953 (cit. on p. 3).

[43] P. Brown, J. Cocke, S. D. Pietra, V. D. Pietra, F. Jelinek, R. Mercer, and P. Roossin. "A statistical approach to language translation". In: *Proceedings of the 12th conference on Computational linguistics - Volume 1*. COLING '88. USA: Association for Computational Linguistics, Aug. 1988, pp. 71–76. DOI: 10.3115/991635.991651 (cit. on p. 50).

[44] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. "Language Models are Few-Shot Learners". In: *arXiv:2005.14165 [cs]* (July 2020) (cit. on p. 41).

[45] F. Burlot, M. García-Martínez, L. Barrault, F. Bougares, and F. Yvon. "Word Representations in Factored Neural Machine Translation". In: *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 20–31. DOI: 10.18653/v1/W17-4703 (cit. on p. 53).

[46] K. Bush. "Aerial". In: *Aerial Tal* (2005) (cit. on p. 49).

[47] L. Caroll. *Alice's Adventures in Wonderland*. 1865 (cit. on pp. xix, 11).

[48] R. Caruana, S. Lawrence, and C. Giles. "Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping". In: *Advances in Neural Information Processing Systems*. Ed. by T. Leen, T. Dietterich, and V. Tresp. Vol. 13. MIT Press, 2001 (cit. on p. 41).

[49] N. Casas, C. Escolano, M. R. Costa-jussà, and J. A. R. Fonollosa. "The TALP-UPC Machine Translation Systems for WMT18 News Shared Translation Task". In: *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 355–360. DOI: 10.18653/v1/W18-6406 (cit. on p. 52).

[50] N. Casas, J. A. R. Fonollosa, C. Escolano, C. Basta, and M. R. Costa-jussà. "The TALP-UPC Machine Translation Systems for WMT19 News Translation Task: Pivoting Techniques for Low Resource MT". In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 155–162. DOI: 10.18653/v1/W19-5311 (cit. on p. 52).

[51] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (June 2002), pp. 321–357. DOI: 10.1613/jair.953 (cit. on p. 38).

[52] W. Che, Z. Yu, Z. Yu, Y. Wen, and J. Guo. "Towards Integrated Classification Lexicon for Handling Unknown Words in Chinese-Vietnamese Neural Machine Translation". In: *ACM Transactions on Asian and Low-Resource Language Information Processing* 19.3 (Apr. 2020), 42:1–42:17. DOI: 10.1145/3373267 (cit. on p. 129).

[53] H. Chen, S. Huang, D. Chiang, X. Dai, and J. Chen. "Combining Character and Word Information in Neural Machine Translation Using a Multi-Level Attention". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1284–1293. DOI: 10.18653/v1/N18-1116 (cit. on p. 53).

[54] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179 (cit. on pp. 27–29).

[55] A. M. Ciobanu and L. P. Dinu. "Automatic Detection of Cognates Using Orthographic Alignment". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 99–105. DOI: 10.3115/v1/P14-2017 (cit. on p. 16).

[56] A. M. Ciobanu and L. P. Dinu. "Automatic Discrimination between Cognates and Borrowings". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 431–437. DOI: 10.3115/v1/P15-2071 (cit. on pp. 16, 17, 147).

[57] A. M. Ciobanu and L. P. Dinu. "Automatic Identification and Production of Related Words for Historical Linguistics". In: *Computational Linguistics* 45.4 (Dec. 2019), pp. 667–704. DOI: 10.1162/coli_a_00361 (cit. on pp. 16, 147).

[58] A. M. Ciobanu and L. P. Dinu. "Automatic Identification and Production of Related Words for Historical Linguistics". In: *Computational Linguistics* 45.4 (Jan. 2020), pp. 667–704. DOI: 10.1162/coli_a_00361 (cit. on pp. 4, 16).

[59] A. M. Ciobanu, L. P. Dinu, and L. Zoicas. "Automatic Reconstruction of Missing Romanian Cognates and Unattested Latin Words". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 3226–3231 (cit. on p. 16).

[60] P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. L. de Hoon. "Biopython: freely available Python tools for computational molecular biology and bioinformatics". In: *Bioinformatics* 25.11 (Mar. 2009), pp. 1422–1423. DOI: 10.1093/bioinformatics/btp163 (cit. on p. 108).

[61] G.-L. Coeurdoux. "Supplément au mémoire qui précède". In: *Mémoire de littérature, tirés des régistres de l'Academie Royale des Inscriptions et Belles-Lettres, depuis l'année*. Imprimerie Impériale, 1768, pp. 647–667 (cit. on p. 3).

[62] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni. "What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 2126–2136. DOI: 10.18653/v1/P18-1198 (cit. on pp. 4, 46, 100, 123).

[63] A. Cser. "Aspects of the phonology and morphology of Classical Latin". PhD thesis. Pázmány Péter Katolikus Egyetem, 2016. Chap. 3 (cit. on p. 64).

[64] O. Čulo and J. Nitzke. "Patterns of Terminological Variation in Post-editing and of Cognate Use in Machine Translation in Contrast to Human Translation". In: *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*. 2016, pp. 106–114 (cit. on p. 15).

[65] A. Currey, A. V. Miceli Barone, and K. Heafield. "Copied Monolingual Data Improves Low-Resource Neural Machine Translation". In: *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 148–156. DOI: 10.18653/v1/W17-4715 (cit. on p. 52).

[66] H. Daumé III and J. Jagarlamudi. "Domain Adaptation for Machine Translation by Mining Unseen Words". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 407–412 (cit. on p. 42).

[67] F. De Toni, C. Akiki, J. de la Rosa, C. Fourrier, E. Manjavacas, S. Schweter, and D. van Strien. "Entities, Dates, and Languages: Zero-Shot on Historical Texts with T0". In: *Proceedings of the International Workshop on Challenges & Perspectives in Creating Large Language Models 2022 (BigScience 2022)*. Dublin, France, May 2022 (cit. on p. 6).

[68] P. Dekker. "Reconstructing language ancestry by performing word prediction with neural networks". MA thesis. University of Amsterdam, 2018 (cit. on pp. 5, 16–18, 59, 71, 145, 147).

[69] P. Dekker and W. Zuidema. "Word prediction in computational historical linguistics". In: *Journal of Language Modelling* 8.2 (Feb. 2021), pp. 295–336. DOI: 10.15398/jlm.v8i2.268 (cit. on pp. 4, 17).

[70] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv:1810.04805 [cs]* (May 2019) (cit. on pp. 35, 41).

[71] Y. Ding, Y. Liu, H. Luan, and M. Sun. "Visualizing and Understanding Neural Machine Translation". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1150–1159. DOI: 10.18653/v1/P17-1106 (cit. on p. 46).

[72] L. Dinu and A. M. Ciobanu. "Building a Dataset of Multilingual Cognates for the Romanian Lexicon". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 1038–1043 (cit. on p. 16).

[73] M. E. Dorman. "Sound Change Patterns from Latin in the Following Romance Languages: Italian, Spanish, Portuguese, and French". Bachelors. University of Arizona, 2010 (cit. on p. 242).

[74] F. Doshi-Velez and B. Kim. "Towards A Rigorous Science of Interpretable Machine Learning". In: *arXiv:1702.08608 [cs, stat]* (Mar. 2017). arXiv: 1702.08608 (cit. on p. 44).

[75] M. Dowling, T. Lynn, A. Poncelas, and A. Way. "SMT versus NMT: Preliminary comparisons for Irish". In: *Proceedings of the AMTA 2018 Workshop on Technologies for MT of Low Resource Languages (LoResMT 2018)*. Boston, MA: Association for Machine Translation in the Americas, Mar. 2018, pp. 12–20 (cit. on pp. 51, 90).

[76] J. Duddington. *eSpeak text to speech*. 2007-2015 (cit. on p. 65).

[77] L. Duong, H. Kanayama, T. Ma, S. Bird, and T. Cohn. "Learning Crosslingual Word Embeddings without Bilingual Corpora". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1285–1295. DOI: 10.18653/v1/D16-1136 (cit. on pp. 54, 129).

[78] P. Durkin. "Etymology". In: *The Oxford handbook of the word* (2015), pp. 444–458 (cit. on pp. 4, 15, 145, 147).

[79] N. Durrani, F. Dalvi, H. Sajjad, Y. Belinkov, and P. Nakov. "One Size Does Not Fit All: Comparing NMT Representations of Different Granularities". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 1504–1516. DOI: 10.18653/v1/N19-1154 (cit. on pp. 39, 42).

[80] S. Edunov, M. Ott, M. Auli, and D. Grangier. "Understanding Back-Translation at Scale". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 489–500. DOI: 10.18653/v1/D18-1045 (cit. on p. 52).

[81] S. H. Elgin. "For the Sake of Grace". In: *The Magazine of Fantasy and Science Fiction* (May 1969) (cit. on p. 71).

[82] J. L. Elman. "Finding structure in time". In: *Cognitive Science* 14.2 (1990), pp. 179–211. DOI: https://doi.org/10.1016/0364-0213(90)90002-E (cit. on p. 25).

[83] M. Elsner, A. D. Sims, A. Erdmann, A. Hernandez, E. Jaffe, L. Jin, M. B. Johnson, S. Karim, D. L. King, L. L. Nunes, B.-D. Oh, N. Rasmussen, C. Shain, S. Antetomaso, K. V. Dickinson, N. Diewald, M. McKenzie, and S. Stevens-Guille. "Modeling morphological learning, typology, and change: What can the neural sequence-to-sequence framework contribute?" In: *Journal of Language Modelling* 7.1 (Dec. 2019), pp. 53–98-53–98. DOI: 10.15398/jlm.v7i1.244 (cit. on p. 44).

[84] M. Fadaee, A. Bisazza, and C. Monz. "Data Augmentation for Low-Resource Neural Machine Translation". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 567–573. DOI: 10.18653/v1/P17-2090 (cit. on p. 52).

[85] M. Fadaee and C. Monz. "Back-Translation Sampling by Targeting Difficult Words in Neural Machine Translation". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 436–446. DOI: 10.18653/v1/D18-1040 (cit. on p. 52).

[86] F.-L. Fan, J. Xiong, M. Li, and G. Wang. "On Interpretability of Artificial Neural Networks: A Survey". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* (2021). DOI: 10.1109/TRPMS.2021.3066428 (cit. on p. 44).

[87] O. Firat, K. Cho, and Y. Bengio. "Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 866–875. DOI: 10.18653/v1/N16-1101 (cit. on p. 50).

[88] O. Firat, K. Cho, and Y. Bengio. "Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 866–875. DOI: 10.18653/v1/N16-1101 (cit. on p. 53).

[89] K. Fort, G. Adda, and K. B. Cohen. "Last Words: Amazon Mechanical Turk: Gold Mine or Coal Mine?" In: *Computational Linguistics* 37.2 (June 2011), pp. 413–420. DOI: 10.1162/COLI_a_00057 (cit. on p. 45).

[90] B. W. Fortson IV. *Indo-European language and culture: An introduction*. Vol. 30. John Wiley & Sons, 2011 (cit. on pp. 14, 15).

[91] C. Fourrier. "Évolution phonologique des langues et réseaux de neurones : travaux préliminaires (Sound change and neural networks: preliminary experiments)". French. In: *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL*. Nancy, France: ATALA et AFCP, June 2020, pp. 110–122 (cit. on pp. 59, 63, 73, 146).

[92] C. Fourrier, R. Bawden, and B. Sagot. "Can Cognate Prediction Be Modelled as a Low-Resource Machine Translation Task?" In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 847–861. DOI: 10.18653/v1/2021.findings-acl.75 (cit. on pp. 6, 17, 83, 100, 146).

[93] C. Fourrier and S. Montariol. "Caveats of Measuring Semantic Change of Cognates and Borrowings using Multilingual Word Embeddings". In: *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*. May 2022 (cit. on pp. 6, 148).

[94] C. Fourrier and B. Sagot. "Probing Multilingual Cognate Prediction Models". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland, May 2022 (cit. on pp. 17, 146).

[95] C. Fourrier and B. Sagot. "Comparing Statistical and Neural Models for Learning Sound Correspondences". English. In: *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 79–83 (cit. on pp. 63, 73, 146).

[96] C. Fourrier and B. Sagot. "Methodological Aspects of Developing and Managing an Etymological Lexical Resource: Introducing EtymDB-2.0". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 3207–3216 (cit. on pp. 6, 17, 59, 65, 146).

[97] E. Frossard, M. Coustaty, A. Doucet, A. Jatowt, and S. Hengchen. "Dataset for Temporal Analysis of English-French Cognates". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 855–859 (cit. on p. 16).

[98] O. Frunza and D. Inkpen. "Semi-Supervised Learning of Partial Cognates Using Bilingual Bootstrapping". In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 441–448. DOI: 10.3115/1220175.1220231 (cit. on pp. 14–16, 145).

[99] O. Frunza and D. Inkpen. "Identification and disambiguation of cognates, false friends, and partial cognates using machine learning techniques". In: *International Journal of Linguistics* 1.1 (2009), pp. 1–37 (cit. on pp. 4, 14–16).

[100] O. M. Frunza. "Automatic Identification of Cognates, False Friends, and Partial Cognates". Master of Science. 2006 (cit. on pp. 15, 16, 145).

[101] V. Gautam, W. Y. Li, Z. Mahmood, F. Mailhot, S. Nadig, R. Wang, and N. Zhang. "Avengers, Ensemble! Benefits of ensembling in grapheme-to-phoneme prediction". In: *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Online: Association for Computational Linguistics, Aug. 2021, pp. 141–147. DOI: 10.18653/v1/2021.sigmorphon-1.16 (cit. on p. 42).

[102] M. Geva, Y. Goldberg, and J. Berant. "Are We Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 1161–1166. DOI: 10.18653/v1/D19-1107 (cit. on p. 38).

[103] M. Giulianelli, J. Harding, F. Mohnert, D. Hupkes, and W. Zuidema. "Under the Hood: Using Diagnostic Classifiers to Investigate and Improve how Language Models Track Agreement Information". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 240–248. DOI: 10.18653/v1/W18-5426 (cit. on p. 46).

[104] R. van der Goot. "We Need to Talk About train-dev-test Splits". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4485–4494. DOI: 10.18653/v1/2021.emnlp-main.368 (cit. on p. 39).

[105] O. Gotoh. "An improved algorithm for matching biological sequences". In: *Journal of Molecular Biology* 162.3 (1982), pp. 705–708. DOI: https://doi.org/10.1016/0022-2836(82)90398-9 (cit. on p. 108).

[106] S. Gouws and A. Søgaard. "Simple task-specific bilingual word embeddings". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May 2015, pp. 1386–1390. DOI: 10.3115/v1/N15-1157 (cit. on p. 54).

[107] S. J. Greenhill. "Levenshtein Distances Fail to Identify Language Relationships Accurately". In: *Computational Linguistics* 37.4 (Dec. 2011), pp. 689–698. DOI: 10.1162/COLI_a_00073 (cit. on pp. 16, 17, 147).

[108] S.-A. Grönroos, S. Virpioja, and M. Kurimo. "Cognate-aware morphological segmentation for multilingual neural translation". In: *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. Belgium, Brussels: Association for Computational Linguistics, 2018, pp. 386–393. DOI: 10.18653/v1/W18-6410 (cit. on p. 4).

[109] J. Gu, H. Hassan, J. Devlin, and V. O. Li. "Universal Neural Machine Translation for Extremely Low Resource Languages". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 344–354. DOI: 10.18653/v1/N18-1032 (cit. on p. 41).

[110] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio. "On Using Monolingual Corpora in Neural Machine Translation". In: *arXiv:1503.03535 [cs]* (June 2015) (cit. on p. 53).

[111] J. Guo, L. Xu, and E. Chen. "Jointly Masked Sequence-to-Sequence Model for Non-Autoregressive Neural Machine Translation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 376–385. DOI: 10.18653/v1/2020.acl-main.36 (cit. on p. 53).

[112] H. Haddad, H. Fadaei, and H. Faili. "Handling OOV Words in NMT Using Unsupervised Bilingual Embedding". In: *2018 9th International Symposium on Telecommunications (IST)*. Dec. 2018, pp. 569–574. DOI: 10.1109/IS.2018.8661016 (cit. on p. 54).

[113] B. Haddow, R. Bawden, A. V. M. Barone, J. Helcl, and A. Birch. "Survey of Low-Resource Machine Translation". In: *arXiv:2109.00486 [cs]* (Sept. 2021). arXiv: 2109.00486 (cit. on p. 51).

[114] M. Hahn and M. Baroni. "Tabula Nearly Rasa: Probing the Linguistic Knowledge of Character-level Neural Language Models Trained on Unsegmented Text". In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 467–484. DOI: 10.1162/tacl_a_00283 (cit. on p. 4).

[115] D. Hall and D. Klein. "Finding Cognate Groups Using Phylogenies". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 1030–1039 (cit. on pp. 16, 17).

[116] D. Hall and D. Klein. "Large-Scale Cognate Recovery". In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, July 2011, pp. 344–354 (cit. on pp. 16, 17).

[117] M. Hämäläinen and J. Rueter. "Finding Sami Cognates with a Character-Based NMT Approach". In: *Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers)*. Honolulu: Association for Computational Linguistics, Feb. 2019, pp. 39–45 (cit. on pp. 14, 16, 17).

[118] V. Hanoka and B. Sagot. "An Open-Source Heavily Multilingual Translation Graph Extracted from Wiktionaries and Parallel Corpora". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 3179–3186 (cit. on p. 66).

[119] A. Hantgan and J.-M. List. "Bangime: Secret Language, Language Isolate, or Language Island?" 2018 (cit. on p. 147).

[120] D. Harbecke, R. Schwarzenberg, and C. Alt. "Learning Explanations from Language Data". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 316–318. DOI: 10.18653/v1/W18-5434 (cit. on p. 46).

[121] F. Hartmann. "Predicting Historical Phonetic Features using Deep Neural Networks: A Case Study of the Phonetic System of Proto-Indo-European". In: *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 98–108. DOI: 10.18653/v1/W19-4713 (cit. on p. 16).

[122] Hasigaowa and S. Wang. "Research on Unknown Words Processing of Mongolian-Chinese Neural Machine Translation Based on Semantic Similarity". In: *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*. Feb. 2019, pp. 370–374. DOI: 10.1109/CCOMS.2019.8821725 (cit. on p. 52).

[123] B. Hauer, A. A. Habibi, Y. Luan, R. R. Riyadh, and G. Kondrak. "Cognate Projection for Low-Resource Inflection Generation". In: *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 6–11. DOI: 10.18653/v1/W19-4202 (cit. on p. 17).

[124] B. Hauer and G. Kondrak. "Clustering Semantically Equivalent Words into Cognate Sets in Multilingual Lists". In: *Proceedings of 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, Nov. 2011, pp. 865–873 (cit. on pp. 4, 14, 16).

[125] B. Hauer, G. Nicolai, and G. Kondrak. "Bootstrapping Unsupervised Bilingual Lexicon Induction". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 619–624 (cit. on pp. 15, 17, 148).

[126] T. He, J. Chen, X. Tan, and T. Qin. "Language Graph Distillation for Low-Resource Machine Translation". In: *arXiv:1908.06258 [cs]* (Aug. 2019) (cit. on p. 54).

[127] K. Heafield. "KenLM: Faster and Smaller Language Model Queries". In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland, UK: Association for Computational Linguistics, 2011, pp. 187–197 (cit. on p. 61).

[128] W. Heeringa and B. Joseph. "The Relative Divergence of Dutch Dialect Pronunciations from their Common Source: An Exploratory Study". In: *Proceedings of Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 31–39 (cit. on p. 16).

[129] J. Hewitt and P. Liang. "Designing and Interpreting Probes with Control Tasks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2733–2743. DOI: 10.18653/v1/D19-1275 (cit. on p. 123).

[130] J. Hewson. "Reconstructing Prehistoric Languages on the Computer: The Triumph of the Electronic Neogrammarian". In: *COLING 1973 Volume 1: Computational And Mathematical Linguistics: Proceedings of the International Conference on Computational Linguistics*. 1973 (cit. on pp. 16, 145, 147).

[131] V. C. D. Hoang, P. Koehn, G. Haffari, and T. Cohn. "Iterative Back-Translation for Neural Machine Translation". In: *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 18–24. DOI: 10.18653/v1/W18-2703 (cit. on p. 52).

[132] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735 (cit. on p. 26).

[133] P.-E. Honnet, A. Popescu-Belis, C. Musat, and M. Baeriswyl. "Machine Translation of Low-Resource Spoken Dialects: Strategies for Normalizing Swiss German". In: *arXiv:1710.11035 [cs]* (Feb. 2018) (cit. on p. 131).

[134] K. Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural Networks* 4.2 (Jan. 1991), pp. 251–257. DOI: 10.1016/0893-6080(91)90009-T (cit. on p. 22).

[135] K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (Jan. 1989), pp. 359–366. DOI: 10.1016/0893-6080(89)90020-8 (cit. on p. 22).

[136] D. Hovy and C. Purschke. "Capturing Regional Variation with Distributed Place Representations and Geographic Retrofitting". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 4383–4394. DOI: 10.18653/v1/D18-1469 (cit. on p. 148).

[137] M. Hu, Y. Wu, S. Zhao, H. Guo, R. Cheng, and Z. Su. "Domain-Invariant Feature Distillation for Cross-Domain Sentiment Classification". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5559–5568. DOI: 10.18653/v1/D19-1558 (cit. on pp. 52, 129).

[138] D. Huck, A. Bothorel-Witz, and A. Geiger-Jaillet. *L'Alsace et ses langues. Eléments de description d'une situation sociolinguistique en zone frontalière*. Language Bridges, a Working Group of the Interreg IIIC project 'Change on Borders', 2007 (cit. on p. 129).

[139] A. Hudlett. "La Charte de la graphie harmonisée des parlers alsaciens ou Comment gérer la variation diatopique de la graphie de l'alsacien?" In: *Nouveaux cahiers d'allemand* 22.1 (2004), pp. 103–109 (cit. on p. 131).

[140] D. Inkpen, O. Frunza, and G. Kondrak. "Automatic Identification of Cognates and False Friends in French and English". In: *Proceedings of Recent Advances in Natural Language Processing 2005*. Borovets, 2005, pp. 251–257 (cit. on pp. 15, 16).

[141] International Phonetic Association. *Handbook of the International Phonetic Association : A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press, 1999 (cit. on p. 11).

[142] A. Irvine and C. Callison-Burch. "A Comprehensive Analysis of Bilingual Lexicon Induction". In: *Computational Linguistics* 43.2 (June 2017), pp. 273–310. DOI: 10.1162/COLI_a_00284 (cit. on p. 131).

[143] A. Jacovi and Y. Goldberg. "Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness?" In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4198–4205. DOI: 10.18653/v1/2020.acl-main.386 (cit. on pp. 45, 112).

[144] G. Jäger and J.-M. List. "Statistical and computational elaborations of the classical comparative method". 2016 (cit. on pp. 16, 147).

[145] G. Jäger, J.-M. List, and P. Sofroniev. "Using support vector machines and state-of-the-art algorithms for phonetic alignment to identify cognates in multi-lingual wordlists". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 1205–1216 (cit. on p. 16).

[146] Z. Jamil, D. Inkpen, P. Buddhitha, and K. White. "Monitoring Tweets for Depression to Detect At-risk Users". In: *Proceedings of the Fourth Workshop on Computational Linguistics and Clinical Psychology — From Linguistic Signal to Clinical Reality*. Vancouver, BC: Association for Computational Linguistics, Aug. 2017, pp. 32–40. DOI: 10.18653/v1/W17-3104 (cit. on p. 38).

[147] N. Japkowicz. "The Class Imbalance Problem: Significance and Strategies". In: *Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI*. 2000, pp. 111–117 (cit. on p. 38).

[148] G. Jawahar, B. Sagot, and D. Seddah. "What Does BERT Learn about the Structure of Language?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 3651–3657. DOI: 10.18653/v1/P19-1356 (cit. on p. 46).

[149] S. Jean, K. Cho, R. Memisevic, and Y. Bengio. "On Using Very Large Target Vocabulary for Neural Machine Translation". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 1–10. DOI: 10.3115/v1/P15-1001 (cit. on p. 42).

[150] J.-y. Jo and S.-H. Myaeng. "Roles and Utilization of Attention Heads in Transformer-based Neural Language Models". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 3404–3417. DOI: 10.18653/v1/2020.acl-main.311 (cit. on p. 4).

[151] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean. "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation". In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 339–351. DOI: 10.1162/tacl_a_00065 (cit. on pp. 50, 53).

[152] W. Jones. *The Third Anniversary Discourse*. Delivered 2 February, 1786 by the President, at the Asiatick Society of Bengal "On the Hindús". 1786 (cit. on p. 3).

[153] D. Kanojia, S. Munukutla, S. Ghodekar, P. Bhattacharyya, and M. Kulkarni. "Keep Your Dimensions on a Leash: True Cognate Detection using Siamese Deep Neural Networks". In: *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*. CoDS COMAD 2020. New York, NY, USA: Association for Computing Machinery, Jan. 2020, pp. 324–325. DOI: 10.1145/3371158.3371207 (cit. on pp. 15, 16).

[154] D. Kanojia, K. Patel, P. Bhattacharyya, M. Kulkarni, and G. Haffari. "Utilizing Wordnets for Cognate Detection among Indian Languages". In: Dec. 2021 (cit. on p. 15).

[155] D. Kanojia, P. Sharma, S. Ghodekar, P. Bhattacharyya, G. Haffari, and M. Kulkarni. "Cognition-aware Cognate Detection". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 3281–3292. DOI: 10.18653/v1/2021.eacl-main.288 (cit. on pp. 15, 16).

[156] J. Khatri and P. Bhattacharyya. "Utilizing Monolingual Data in NMT for Similar Languages: Submission to Similar Language Translation Task". In: *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 197–201. DOI: 10.18653/v1/W19-5426 (cit. on p. 52).

[157] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *ICLR 2015 Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. San Diego, CA, USA, 2015 (cit. on pp. 42, 62).

[158] T. Kocmi and O. Bojar. "Trivial Transfer Learning for Low-Resource Neural Machine Translation". In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 244–252. DOI: 10.18653/v1/W18-6325 (cit. on p. 52).

[159] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. "Moses: Open source toolkit for statistical machine translation". In: *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*. 2007, pp. 177–180. DOI: 10.5555/1557769.1557821 (cit. on pp. 50, 61).

[160] P. Koehn, H. Khayrallah, K. Heafield, and M. L. Forcada. "Findings of the WMT 2018 Shared Task on Parallel Corpus Filtering". In: *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 726–739. DOI: 10.18653/v1/W18-6453 (cit. on p. 39).

[161] G. Kondrak. "Identifying Cognates by Phonetic and Semantic Similarity". In: *Second Meeting of the North American Chapter of the Association for Computational Linguistics*. 2001 (cit. on pp. 15, 16).

[162] G. Kondrak. "Determining Recurrent Sound Correspondences by Inducing Translation Models". In: *COLING 2002: The 19th International Conference on Computational Linguistics*. 2002 (cit. on p. 16).

[163] G. Kondrak, D. Marcu, and K. Knight. "Cognates Can Improve Statistical Translation Models". In: *Companion Volume of the Proceedings of HLT-NAACL 2003 - Short Papers*. 2003, pp. 46–48 (cit. on pp. 15, 16).

[164] M. A. Kramer. "Nonlinear principal component analysis using autoassociative neural networks". In: *AIChE journal* 37.2 (1991), pp. 233–243 (cit. on p. 35).

[165] T. Krefeld. "Cognitive ease and lexical borrowing: the recategorization of body parts in Romance". In: *Cognitive ease and lexical borrowing: the recategorization of body parts in Romance*. De Gruyter Mouton, Mar. 2013, pp. 259–278. DOI: 10.1515/9783110804195.259 (cit. on p. 15).

[166] J. Kreutzer, I. Caswell, L. Wang, A. Wahab, D. van Esch, N. Ulzii-Orshikh, A. Tapo, N. Subramani, A. Sokolov, C. Sikasote, M. Setyawan, S. Sarin, S. Samb, B. Sagot, C. Rivera, A. Rios, I. Papadimitriou, S. Osei, P. Ortiz Suarez, I. Orife, K. Ogueji, R. A. Niyongabo, T. Q. Nguyen, M. Müller, A. Müller, S. H. Muhammad, N. Muhammad, A. Mnyakeni, J. Mirzakhalov, T. Matangira, C. Leong, N. Lawson, S. Kudugunta, Y. Jernite, M. Jenny, O. Firat, B. F. P. Dossou, S. Dlamini, N. de Silva, S. Ç. Ballı, S. Biderman, A. Battisti, A. Baruwa, A. Bapna, P. Baljekar, I. A. Azime, A. Awokoya, D. Ataman, O. Ahia, O. Ahia, S. Agrawal, and M. Adeyemi. "Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets". In: *Transactions of the Association for Computational Linguistics* 10 (2022), pp. 50–72. DOI: 10.1162/tacl_a_00447 (cit. on p. 37).

[167] M. Kubat and S. Matwin. "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection". In: *Proceedings of the Fourteenth International Conference on Machine Learning*. Morgan Kaufmann, 1997, pp. 179–186 (cit. on p. 38).

[168] A. Kuhnle and A. Copestake. "Deep learning evaluation using deep linguistic processing". In: *Proceedings of the Workshop on Generalization in the Age of Deep Learning*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 17–23. DOI: 10.18653/v1/W18-1003 (cit. on p. 45).

[169] S. Kumar, A. Vaidya, and S. Agarwal. "Discovering Cognates Using LSTM Networks". In: *4th Annual Conference of the Association for Cognitive Science*. Hyderabad, India, 2017 (cit. on pp. 16, 145).

[170] P. Ladefoged. "The IPA and a theory of phonetic description". In: *Proceedings of the KSPS conference*. The Korean Society Of Phonetic Sciences and Speech Technology. 1996, pp. 27–36 (cit. on p. 11).

[171] R. A. Lafferty. "About a Secret Crocodile". In: *Does Anyone Else Have Something Further to Add?* (May 1974) (cit. on p. 83).

[172] V. Laippala, S. Rönnqvist, S. Hellström, J. Luotolahti, L. Repo, A. Salmela, V. Skantsi, and S. Pyysalo. "From Web Crawl to Clean Register-Annotated Corpora". English. In: *Proceedings of the 12th Web as Corpus Workshop*. Marseille, France: European Language Resources Association, May 2020, pp. 14–22 (cit. on p. 39).

[173] G. Lample and A. Conneau. "Cross-lingual Language Model Pretraining". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2019) (cit. on p. 53).

[174] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. "Unsupervised Machine Translation Using Monolingual Corpora Only". In: *ICLR 2017 Conference Track Proceedings*. Oct. 2017 (cit. on p. 52).

[175] J. Laurikkala. "Improving Identification of Difficult Small Classes by Balancing Class Distribution". In: *Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine*. AIME '01. Berlin, Heidelberg: Springer-Verlag, July 2001, pp. 63–66 (cit. on p. 38).

[176] A. Lauscher, V. Ravishankar, I. Vulić, and G. Glavaš. "From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4483–4499. DOI: 10.18653/v1/2020.emnlp-main.363 (cit. on pp. 41, 52).

[177] E. Lefever, S. Labat, and P. Singh. "Identifying Cognates in English-Dutch and French-Dutch by means of Orthographic Information and Cross-lingual Word Embeddings". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 4096–4101 (cit. on pp. 15, 16).

[178] O. Levy, Y. Goldberg, and I. Dagan. "Improving Distributional Similarity with Lessons Learned from Word Embeddings". In: *Transactions of the Association for Computational Linguistics* 3 (2015), pp. 211–225. DOI: 10.1162/tacl_a_00134 (cit. on p. 42).

[179] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *arXiv:1910.13461 [cs, stat]* (Oct. 2019) (cit. on p. 41).

[180] B. Li, H. Liu, Z. Wang, Y. Jiang, T. Xiao, J. Zhu, T. Liu, and C. Li. "Does Multi-Encoder Help? A Case Study on Context-Aware Neural Machine Translation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 3512–3518. DOI: 10.18653/v1/2020.acl-main.322 (cit. on pp. 50, 52).

[181] X. Li, J. Zhang, and C. Zong. "Towards zero unknown word in neural machine translation". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI'16. New York, New York, USA: AAAI Press, July 2016, pp. 2852–2858 (cit. on pp. 54, 129).

[182] X. Li, H. Xiong, X. Li, X. Wu, X. Zhang, J. Liu, J. Bian, and D. Dou. "Interpretable Deep Learning: Interpretation, Interpretability, Trustworthiness, and Beyond". In: *arXiv:2103.10689 [cs]* (May 2021) (cit. on p. 44).

[183] Y. Li and C. Scarton. "Revisiting Rumour Stance Classification: Dealing with Imbalanced Data". In: *Proceedings of the 3rd International Workshop on Rumours and Deception in Social Media (RDSM)*. Barcelona, Spain (Online): Association for Computational Linguistics, Dec. 2020, pp. 38–44 (cit. on p. 38).

[184] Z. Li and L. Specia. "A Comparison on Fine-grained Pre-trained Embeddings for the WMT19Chinese-English News Translation Task". In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 249–256. DOI: 10.18653/v1/W19-5324 (cit. on p. 42).

[185] Z. Li and L. Specia. "Improving Neural Machine Translation Robustness via Data Augmentation: Beyond Back-Translation". In: *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 328–336. DOI: 10.18653/v1/D19-5543 (cit. on p. 52).

[186] Z. Li, H. Zhao, R. Wang, M. Utiyama, and E. Sumita. "Reference Language based Unsupervised Neural Machine Translation". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4151–4162. DOI: 10.18653/v1/2020.findings-emnlp.371 (cit. on p. 52).

[187] J. Libovický, V. Hangya, H. Schmid, and A. Fraser. "The LMU Munich System for the WMT20 Very Low Resource Supervised MT Task". In: *Proceedings of the Fifth Conference on Machine Translation*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1104–1111 (cit. on pp. 52, 53).

[188] M. Lichouri and M. Abbas. "Simple vs Oversampling-based Classification Methods for Fine Grained Arabic Dialect Identification in Twitter". In: *Proceedings of the Fifth Arabic Natural Language Processing Workshop*. Barcelona, Spain (Online): Association for Computational Linguistics, Dec. 2020, pp. 250–256 (cit. on p. 38).

[189] C.-Y. Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81 (cit. on p. 50).

[190] J.-M. List. "LexStat: Automatic Detection of Cognates in Multilingual Wordlists". In: *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*. Avignon, France: Association for Computational Linguistics, Apr. 2012, pp. 117–125 (cit. on p. 16).

[191] J.-M. List and R. Forkel. "Automated identification of borrowings in multilingual wordlists [version 3; peer review: 4 approved]". In: *Open Research Europe* 1.79 (2022). DOI: 10.12688/openreseurope.13843.3 (cit. on pp. 16, 147).

[192] J.-M. List, R. Forkel, and N. W. Hill. "A New Framework for Fast Automated Phonological Reconstruction Using Trimmed Alignments and Sound Correspondence Patterns". In: *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*. Apr. 2022 (cit. on pp. 15–17).

[193] J.-M. List, S. J. Greenhill, and R. D. Gray. "The Potential of Automatic Word Comparison for Historical Linguistics". In: *PLOS ONE* 12.1 (2017), pp. 1–18. DOI: 10.1371/journal.pone.0170046 (cit. on pp. 4, 14, 16).

[194] J.-M. List, M. Walworth, S. J. Greenhill, T. Tresoldi, and R. Forkel. "Sequence comparison in computational historical linguistics". In: *Journal of Language Evolution* 3.2 (2018), pp. 130–144. DOI: 10.1093/jole/lzy006. eprint: http://oup.prod.sis.lan/jole/article-pdf/3/2/130/27255915/lzy006.pdf (cit. on p. 16).

[195] C. Liu, Q. Zhang, X. Zhang, K. Singh, Y. Saraf, and G. Zweig. "Multilingual Graphemic Hybrid ASR with Massive Data Augmentation". English. In: *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*. Marseille, France: European Language Resources association, May 2020, pp. 46–52 (cit. on p. 53).

[196] X. Liu and C. Wang. "An Empirical Study on Hyperparameter Optimization for Fine-Tuning Pre-trained Language Models". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 2286–2300. DOI: 10.18653/v1/2021.acl-long.178 (cit. on p. 42).

[197] A. Lovelace. *Correspondences with A. De Morgan*. 1840 (cit. on p. 19).

[198] G. Luo, Y. Yang, Y. Yuan, Z. Chen, and A. Ainiwaer. "Hierarchical Transfer Learning Architecture for Low-Resource Neural Machine Translation". In: *IEEE Access* 7 (2019), pp. 154157–154166. DOI: 10.1109/ACCESS.2019.2936002 (cit. on p. 52).

[199] J. Luo, F. Hartmann, E. Santus, R. Barzilay, and Y. Cao. "Deciphering Undersegmented Ancient Scripts Using Phonetic Prior". In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 69–81. DOI: 10.1162/tacl_a_00354 (cit. on pp. 16, 17, 148).

[200] M.-T. Luong and C. D. Manning. "Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1054–1063. DOI: 10.18653/v1/P16-1100 (cit. on p. 53).

[201] T. Luong, H. Pham, and C. D. Manning. "Effective Approaches to Attention-based Neural Machine Translation". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 1412–1421. DOI: 10.18653/v1/D15-1166 (cit. on pp. 30, 31, 33, 34, 50, 61, 87, 89).

[202] T. Luong, I. Sutskever, Q. Le, O. Vinyals, and W. Zaremba. "Addressing the Rare Word Problem in Neural Machine Translation". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 11–19. DOI: 10.3115/v1/P15-1002 (cit. on pp. 53, 129).

[203] L. V. der Maaten and G. Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605 (cit. on pp. 46, 101).

[204] A. Madsen, S. Reddy, and S. Chandar. "Post-hoc Interpretability for Neural NLP: A Survey". In: *arxiv* (Aug. 2021) (cit. on pp. 44, 46, 101, 112).

[205] G. S. Mann and D. Yarowsky. "Multipath Translation Lexicon Induction via Bridge Languages". In: *Second Meeting of the North American Chapter of the Association for Computational Linguistics*. 2001 (cit. on pp. 4, 15, 16, 131, 142, 148).

[206] M. Maragoudakis, K. Kermanidis, A. Garbis, and N. Fakotakis. "Dealing with Imbalanced Data using Bayesian Techniques". In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy: European Language Resources Association (ELRA), May 2006 (cit. on p. 38).

[207] B. Marie, A. Fujita, and R. Rubino. "Scientific Credibility of Machine Translation Research: A Meta-Evaluation of 769 Papers". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 7297–7306. DOI: 10.18653/v1/2021.acl-long.566 (cit. on p. 43).

[208] B. Marie, R. Rubino, and A. Fujita. "Tagged Back-translation Revisited: Why Does It Really Work?" In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 5990–5997. DOI: 10.18653/v1/2020.acl-main.532 (cit. on p. 52).

[209] I. Markov, V. Nastase, and C. Strapparava. "Anglicized Words and Misspelled Cognates in Native Language Identification". In: *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 275–284. DOI: 10.18653/v1/W19-4429 (cit. on pp. 15, 16).

[210] C. Marr and D. R. Mortensen. "Computerized Forward Reconstruction for Analysis in Diachronic Phonology, and Latin to French Reflex Prediction". English. In: *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 28–36 (cit. on p. 16).

[211] M. Martinc, P. Kralj Novak, and S. Pollak. "Leveraging Contextual Embeddings for Detecting Diachronic Semantic Shift". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 4811–4819 (cit. on p. 147).

[212] Y. Marton, C. Callison-Burch, and P. Resnik. "Improved Statistical Machine Translation Using Monolingually-Derived Paraphrases". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, Aug. 2009, pp. 381–390 (cit. on p. 52).

[213] N. Mathur, T. Baldwin, and T. Cohn. "Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4984–4997. DOI: 10.18653/v1/2020.acl-main.448 (cit. on pp. 43, 50).

[214] P. Matthews. *Concise Oxford Dictionary of Linguistics*. 2nd edition. Oxford University Press, 2007 (cit. on p. 13).

[215] R. T. McCoy and R. Frank. "Phonologically Informed Edit Distance Algorithms for Word Alignment with Low-Resource Languages". In: *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*. 2018, pp. 102–112. DOI: 10.7275/R5251GC0 (cit. on pp. 15, 17).

[216] J. P. McCrae, C. Chiarcos, F. Bond, P. Cimiano, T. Declerck, G. de Melo, J. Gracia, S. Hellmann, B. Klimek, S. Moran, P. Osenova, A. Pareja-Lora, and J. Pool. "The Open Linguistics Working Group: Developing the Linguistic Linked Open Data Cloud". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 2435–2441 (cit. on p. 43).

[217] G. de Melo. *EtymWordNet*. 2014 (cit. on p. 17).

[218] C. Meloni, S. Ravfogel, and Y. Goldberg. "Ab Antiquo: Neural Proto-language Reconstruction". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 4460–4473. DOI: 10.18653/v1/2021.naacl-main.353 (cit. on pp. 5, 16, 17, 100, 101, 111, 112, 115, 121, 122, 131, 145, 146).

[219] T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient Estimation of Word Representations in Vector Space". In: *arXiv:1301.3781 [cs]* (Jan. 2013) (cit. on p. 42).

[220] T. Mikolov, Q. V. Le, and I. Sutskever. "Exploiting Similarities among Languages for Machine Translation". In: *arXiv:1309.4168 [cs]* (Sept. 2013) (cit. on pp. 50, 129).

[221] A. Millour. "Myriadisation de ressources linguistiques pour le traitement automatique de langues non standardisées.(Crowdsourcing linguistic resources for natural non-standardised languages processing)." PhD thesis. Sorbonne University, France, 2020 (cit. on pp. 130, 131).

[222] A. Millour, K. Fort, and P. Magistry. "Répliquer et étendre pour l'alsacien "Étiquetage en parties du discours de langues peu dotées par spécialisation des plongements lexicaux" (Replicating and extending for Alsatian : "POS tagging for low-resource languages by adapting word embeddings")". French. In: *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement*

Automatique des Langues (RÉCITAL, 22e édition). 2e atelier Éthique et TRaitemeNt Automatique des Langues (ETeRNAL). Nancy, France: ATALA et AFCP, June 2020, pp. 29–37 (cit. on p. 130).

[223] R. Mitkov, V. Pekar, D. Blagoev, and A. Mulloni. "Methods for extracting and classifying pairs of cognates and false friends". In: *Machine Translation* 21.1 (Mar. 2007), pp. 29–53. DOI: 10.1007/s10590-008-9034-5 (cit. on pp. 4, 15, 16).

[224] S. Montariol and A. Allauzen. "Measure and Evaluation of Semantic Divergence across Two Languages". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1247–1258. DOI: 10.18653/v1/2021.acl-long.100 (cit. on p. 148).

[225] M. Morishita, Y. Oda, G. Neubig, K. Yoshino, K. Sudoh, and S. Nakamura. "An Empirical Study of Mini-Batch Creation Strategies for Neural Machine Translation". In: *Proceedings of the First Workshop on Neural Machine Translation*. Vancouver: Association for Computational Linguistics, Aug. 2017, pp. 61–68. DOI: 10.18653/v1/W17-3208 (cit. on p. 40).

[226] M. Morishita, J. Suzuki, and M. Nagata. "Improving Neural Machine Translation by Incorporating Hierarchical Subword Features". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 618–629 (cit. on p. 53).

[227] B. Muller, A. Anastasopoulos, B. Sagot, and D. Seddah. "When Being Unseen from mBERT is just the Beginning: Handling New Languages With Multilingual Language Models". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 448–462. DOI: 10.18653/v1/2021.naacl-main.38 (cit. on p. 104).

[228] A. Mulloni. "Automatic Prediction of Cognate Orthography Using Support Vector Machines". In: *Proceedings of the ACL 2007 Student Research Workshop*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 25–30 (cit. on pp. 4, 15, 17, 145).

[229] K. Muthuraman, F. Reiss, H. Xu, B. Cutler, and Z. Eichenberger. "Data Cleaning Tools for Token Classification Tasks". In: *Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances*. Online: Association for Computational Linguistics, June 2021, pp. 59–61. DOI: 10.18653/v1/2021.dash-1.10 (cit. on p. 39).

[230] A. H. Nasution, Y. Murakami, and T. Ishida. "Constraint-Based Bilingual Lexicon Induction for Closely Related Languages". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 3291–3298 (cit. on p. 16).

[231] M. Navlea and A. Todiraşcu. "Cognate Identification for a French - Romanian Lexical Alignment System: Empirical Study". In: *Proceedings of the 15th Annual conference of the European Association for Machine Translation*. Leuven, Belgium: European Association for Machine Translation, May 2011 (cit. on pp. 15, 16).

[232] S. B. Needleman and C. D. Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins". In: *Journal of Molecular Biology* 48.3 (1970), pp. 443–453. DOI: https://doi.org/10.1016/0022-2836(70)90057-4 (cit. on pp. 108, 115).

[233] T.-V. Ngo, T.-L. Ha, P.-T. Nguyen, and L.-M. Nguyen. "Overcoming the Rare Word Problem for low-resource language pairs in Neural Machine Translation". In: *Proceedings of the 6th Workshop on Asian Translation*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 207–214. DOI: 10.18653/v1/D19-5228 (cit. on p. 54).

[234] M. Nguyen, G. H. Ngo, and N. Chen. "Multimodal neural pronunciation modeling for spoken languages with logographic origin". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 2916–2922. DOI: 10.18653/v1/D18-1320 (cit. on pp. 15, 17, 145).

[235] R. Nitschke. "Restoring the Sister: Reconstructing a Lexicon from Sister Languages using Neural Machine Translation". In: *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*. Online: Association for Computational Linguistics, June 2021, pp. 122–130. DOI: 10.18653/v1/2021.americasnlp-1.13 (cit. on pp. 16, 17).

[236] X. Niu, P. Mathur, G. Dinu, and Y. Al-Onaizan. "Evaluating Robustness to Input Perturbations for Neural Machine Translation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 8538–8544. DOI: 10.18653/v1/2020.acl-main.755 (cit. on p. 51).

[237] F. J. Och and H. Ney. "A Systematic Comparison of Various Statistical Alignment Models". In: *Computational Linguistics* 29.1 (2003), pp. 19–51. DOI: 10.1162/089120103321337421 (cit. on p. 61).

[238] P. J. Ortiz Suárez, B. Sagot, and L. Romary. "Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures". In: ed. by P. Bański, A. Barbaresi, H. Biber, E. Breiteneder, S. Clematide, M. Kupietz, H. Lüngen, and C. Iliadi. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019. Mannheim: Leibniz-Institut für Deutsche Sprache, 2019, pp. 9–16. DOI: 10.14618/ids-pub-9021 (cit. on p. 130).

[239] "Vorwort". In: *Morphologische Untersuchungen auf dem Gebiete der indogermanischen Sprachen*. Ed. by H. Osthoff and K. Brugmann. Vol. 1. Cambridge Library Collection - Linguistics. Cambridge: Cambridge University Press, 2014, 1878, pp. iii–xx. DOI: 10.1017/CBO9781139600101.001 (cit. on pp. 3, 14, 145).

[240] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli. "fairseq: A Fast, Extensible Toolkit for Sequence Modeling". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 48–53. DOI: 10.18653/v1/N19-4009 (cit. on p. 50).

[241] L. Padgett. "Mimsy Were the Borogoves". In: *Astounding Science Fiction Magazine.* (1943) (cit. on p. 59).

[242] E. Pantaleo, V. W. Anelli, T. Di Noia, and G. Sérasset. "Etytree: A Graphical and Interactive Etymology Dictionary Based on Wiktionary". In: *Proceedings of the 26th International Conference on World Wide Web Companion*. WWW '17 Companion. Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, pp. 1635–1640. DOI: 10.1145/3041021.3053365 (cit. on p. 18).

[243] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: 10.3115/1073083.1073135 (cit. on p. 50).

[244] R. Pascanu, T. Mikolov, and Y. Bengio. "On the difficulty of training Recurrent Neural Networks". In: *arXiv:1211.5063 [cs]* (Feb. 2013) (cit. on p. 26).

[245] B. Paumier, R. Caron, and Comité consultatif pour la promotion des langues régionales de la pluralité linguistique interne. *Redéfinir une politique publique en faveur des langues régionales et de la pluralité linguistique interne*. 2013 (cit. on p. 129).

[246] K. Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720. eprint: https://doi.org/10.1080/14786440109462720 (cit. on pp. 46, 101).

[247] N. Poerner, B. Roth, and H. Schütze. "Interpretable Textual Neuron Representations for NLP". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 325–327. DOI: 10.18653/v1/W18-5437 (cit. on p. 46).

[248] M. Post. "A Call for Clarity in Reporting BLEU Scores". In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 186–191. DOI: 10.18653/v1/W18-6319 (cit. on pp. 50, 62, 94, 95).

[249] T. Pratchett. *I Shall Wear Midnight*. 2010 (cit. on p. 61).

[250] E. Rabinovich, Y. Tsvetkov, and S. Wintner. "Native Language Cognate Effects on Second Language Lexical Choice". In: *Transactions of the Association for Computational Linguistics* 6 (2018), pp. 329–342. DOI: 10.1162/tacl_a_00024 (cit. on p. 16).

[251] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *arXiv:1910.10683 [cs, stat]* (July 2020) (cit. on p. 41).

[252] A. Raganato and J. Tiedemann. "An Analysis of Encoder Representations in Transformer-Based Machine Translation". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 287–297. DOI: 10.18653/v1/W18-5431 (cit. on pp. 4, 46).

[253] T. Rama. "Automatic cognate identification with gap-weighted string subsequences." In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May 2015, pp. 1227–1231. DOI: 10.3115/v1/N15-1130 (cit. on p. 16).

[254] T. Rama. "Siamese Convolutional Networks for Cognate Identification". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 1018–1027 (cit. on p. 4).

[255] T. Rama and J.-M. List. "An Automated Framework for Fast Cognate Detection and Bayesian Phylogenetic Inference in Computational Historical Linguistics". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 6225–6235. DOI: 10.18653/v1/P19-1627 (cit. on p. 16).

[256] T. Rama, J.-M. List, J. Wahle, and G. Jäger. "Are Automatic Methods for Cognate Detection Good Enough for Phylogenetic Reconstruction in Historical Linguistics?" In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 393–400. DOI: 10.18653/v1/N18-2063 (cit. on pp. 16, 147).

[257] G. Ramírez-Sánchez, J. Zaragoza-Bernabeu, M. Bañón, and S. O. Rojas. "Bifixer and Bicleaner: two open-source tools to clean your parallel data". In: *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*. Lisboa, Portugal: European Association for Machine Translation, Nov. 2020, pp. 291–298 (cit. on p. 39).

[258] E. Reiter. "A Structured Review of the Validity of BLEU". In: *Computational Linguistics* 44.3 (Sept. 2018), pp. 393–401. DOI: 10.1162/coli_a_00322 (cit. on pp. 43, 50).

[259] E. Reiter and A. Belz. "An Investigation into the Validity of Some Metrics for Automatically Evaluating Natural Language Generation Systems". In: *Computational Linguistics* 35.4 (Dec. 2009), pp. 529–558. DOI: 10.1162/coli.2009.35.4.35405 (cit. on p. 43).

[260] A. Rogers, T. Baldwin, and K. Leins. "'Just What do You Think You're Doing, Dave?' A Checklist for Responsible Data Use in NLP". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4821–4833. DOI: 10.18653/v1/2021.findings-emnlp.414 (cit. on p. 37).

[261] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65.6 (1958), pp. 386–408. DOI: 10.1037/h0042519 (cit. on p. 19).

[262] S. Ruder. "An overview of gradient descent optimization algorithms". In: *CoRR* abs/1609.04747 (2016). arXiv: 1609.04747 (cit. on p. 42).

[263] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf. "Transfer Learning in Natural Language Processing". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 15–18. DOI: 10.18653/v1/N19-5004 (cit. on p. 41).

[264] P. Ruiz Fabo, D. Bernhard, and C. Werner. "Création d'un corpus FAIR de théâtre en alsacien et normalisation de variétés non-contemporaines". In: *2èmes journées scientifiques du Groupement de Recherche Linguistique Informatique Formelle et de Terrain (LIFT)*. Ed. by T. Poibeau, Y. Parmentier, and E. Schang. Montrouge, France: CNRS, 2020, pp. 34–43 (cit. on p. 130).

[265] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (Oct. 1986), pp. 533–536. DOI: 10.1038/323533a0 (cit. on p. 23).

[266] H. Saadane, O. Benterki, N. Semmar, and C. Fluhr. "Using Arabic Transliteration to Improve Word Alignment from French- Arabic Parallel Corpora". In: *Fourth Workshop on Computational Approaches to Arabic-Script-based Languages*. San Diego, California, USA: Association for Machine Translation in the Americas, Nov. 2012, pp. 38–46 (cit. on pp. 15, 16).

[267] B. Sagot. "Extracting an Etymological Database from Wiktionary". In: *Electronic Lexicography in the 21st century (eLex 2017)*. Leiden, Netherlands, 2017, pp. 716–728 (cit. on p. 17).

[268] T. Samardzic, Y. Scherrer, and E. Glaser. "Normalising orthographic and dialectal variants for the automatic processing of Swiss German". In: Proceedings of the 7th Language and Technology Conference. 2015 (cit. on p. 131).

[269] V. M. Sánchez-Cartagena, J. A. Pérez-Ortiz, and F. Sánchez-Martínez. "The Universitat d'Alacant Submissions to the English-to-Kazakh News Translation Task at WMT 2019". In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 356–363. DOI: 10.18653/v1/W19-5339 (cit. on p. 52).

[270] R. Schäfer. "On Bias-free Crawling and Representative Web Corpora". In: *Proceedings of the 10th Web as Corpus Workshop*. Berlin: Association for Computational Linguistics, Aug. 2016, pp. 99–105. DOI: 10.18653/v1/W16-2612 (cit. on p. 37).

[271] Y. Scherrer. "Transducteurs à fenêtre glissante pour l'induction lexicale". French. In: *Actes de la 15ème conférence sur le Traitement Automatique des Langues Naturelles. REncontres jeunes Chercheurs en Informatique pour le Traitement Automatique des Langues*. Avignon, France: ATALA, June 2008, pp. 70–79 (cit. on p. 15).

[272] Y. Scherrer and O. Rambow. "Natural Language Processing for the Swiss German Dialect Area". In: *Semantic Approaches in Natural Language Processing - Proceedings of the Conference on Natural Language Processing 2010 (KONVENS)*. Ed. by M. Pinkal, I. Rehbein, and A. Schulte im Walde S. & Storrer. Saarbrücken, Germany: Universaar, 2010, pp. 93–102 (cit. on p. 130).

[273] Y. Scherrer and O. Rambow. "Word-Based Dialect Identification with Georeferenced Rules". In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Cambridge, MA: Association for Computational Linguistics, Oct. 2010, pp. 1151–1161 (cit. on p. 131).

[274] Y. Scherrer and B. Sagot. "A language-independent and fully unsupervised approach to lexicon induction and part-of-speech tagging for closely related languages". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 502–508 (cit. on p. 131).

[275] T. Schick and H. Schütze. "It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 2339–2352. DOI: 10.18653/v1/2021.naacl-main.185 (cit. on p. 41).

[276] L. Schmidt, L. Linder, S. Djambazovska, A. Lazaridis, T. Samardžić, and C. Musat. "A Swiss German Dictionary: Variation in Speech and Writing". In: *arXiv:2004.00139 [cs]* (Mar. 2020) (cit. on p. 131).

[277] R. Sennrich, B. Haddow, and A. Birch. "Improving Neural Machine Translation Models with Monolingual Data". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 86–96. DOI: 10.18653/v1/P16-1009 (cit. on p. 52).

[278] R. Sennrich, B. Haddow, and A. Birch. "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162 (cit. on pp. 39, 43).

[279] R. Sennrich and B. Zhang. "Revisiting Low-Resource Neural Machine Translation: A Case Study". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 211–221. DOI: 10.18653/v1/P19-1021 (cit. on pp. 51, 84).

[280] L. Sepúlveda Torres and S. M. Aluísio. "Using machine learning methods to avoid the pitfall of cognates and false friends in Spanish-Portuguese word pairs". In: *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology*. 2011 (cit. on pp. 15, 16, 145).

[281] A. Siddhant, A. Bapna, Y. Cao, O. Firat, M. Chen, S. Kudugunta, N. Arivazhagan, and Y. Wu. "Leveraging Monolingual Data with Self-Supervision for Multilingual Neural Machine Translation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 2827–2835. DOI: 10.18653/v1/2020.acl-main.252 (cit. on p. 53).

[282] P. Sims-Williams. "Mechanising Historical Phonology". In: *Transactions of the Philological Society* 116.3 (2018), pp. 555–573. DOI: 10.1111/1467-968X.12138 (cit. on pp. 16, 17).

[283] T. D. Singh and A. V. Hujon. "Low Resource and Domain Specific English to Khasi SMT and NMT Systems". In: *Proceedings of the 2020 International Conference on Computational Performance Evaluation (ComPE)*. Shillong, India, 2020, pp. 733–737. DOI: 10.1109/ComPE49325.2020.9200059 (cit. on pp. 51, 90).

[284] L. Sitbon, D. Molla, and H. Wang. "Overview of the 2015 ALTA Shared Task: Identifying French Cognates in English Text". In: *Proceedings of the Australasian Language Technology Association Workshop 2015*. Parramatta, Australia, Dec. 2015, pp. 134–137 (cit. on p. 15).

[285] I. Skadiņa and M. Pinnis. "NMT or SMT: Case Study of a Narrow-domain English-Latvian Post-editing Project". In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 373–383 (cit. on pp. 51, 90).

[286] R. N. Smith. "Automatic Simulation of Historical Change". In: *International Conference on Computational Linguistics COLING 1969: Preprint No. 9*. Sånga Säby, Sweden, Sept. 1969 (cit. on pp. 16, 147).

[287] A. Søgaard, S. Ebert, J. Bastings, and K. Filippova. "We Need To Talk About Random Splits". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 1823–1832. DOI: 10.18653/v1/2021.eacl-main.156 (cit. on p. 39).

[288] E. Soisalon-Soininen and M. Granroth-Wilding. "Cross-Family Similarity Learning for Cognate Identification in Low-Resource Languages". In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. Varna, Bulgaria: INCOMA Ltd., Sept. 2019, pp. 1121–1130. DOI: 10.26615/978-954-452-056-4_129 (cit. on p. 16).

[289] S. Soni and K. Roberts. "Evaluation of Dataset Selection for Pre-Training and Fine-Tuning Transformer Language Models for Clinical Question Answering". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 5532–5538 (cit. on p. 37).

[290] A. St Arnaud, D. Beck, and G. Kondrak. "Identifying Cognate Sets Across Dictionaries of Related Languages". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2519–2528. DOI: 10.18653/v1/D17-1267 (cit. on p. 16).

[291] F. Stahlberg. "Neural Machine Translation: A Review". In: *Journal of Artificial Intelligence Research* 69 (Oct. 2020), pp. 343–418. DOI: 10.1613/jair.1.12007 (cit. on pp. 39, 51).

[292] N. Stephenson. *The Diamond Age*. 1995 (cit. on p. 99).

[293] E. Sulem, O. Abend, and A. Rappoport. "BLEU is Not Suitable for the Evaluation of Text Simplification". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 738–744. DOI: 10.18653/v1/D18-1081 (cit. on p. 43).

[294] I. Sutskever, O. Vinyals, and Q. V. Le. "Sequence to Sequence Learning with Neural Networks". In: *arXiv* (Sept. 2014) (cit. on pp. 28, 29).

[295] M. Swadesh. "Salish Internal Relationships". In: *International Journal of American Linguistics* 16.4 (1950), pp. 157–167 (cit. on p. 4).

[296] M. Swadesh. "Towards Greater Accuracy in Lexicostatistic Dating". In: *International Journal of American Linguistics* 21.2 (1955), pp. 121–137 (cit. on pp. 4, 17, 59).

[297] G. Tang, R. Sennrich, and J. Nivre. "Understanding Pure Character-Based Neural Machine Translation: The Case of Translating Finnish into English". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 4251–4262. DOI: 10.18653/v1/2020.coling-main.375 (cit. on p. 45).

[298] A. Towles. *A Gentleman in Moscow*. 2016 (cit. on p. 129).

[299] M. Towsey, J. Diederich, I. Schellhammer, S. Chalup, and C. Brugman. "Natural Language Learning by Recurrent Neural Networks: A Comparison wRh probabilistic approaches". In: *New Methods in Language Processing and Computational Natural Language Learning*. 1998 (cit. on p. 25).

[300] A. Turing. "Intelligent Machinery". 1948 (cit. on p. 19).

[301] A. S. Uban, A. M. Cristea, A. Dinu, L. P. Dinu, S. Georgescu, and L. Zoicas. "Tracking Semantic Change in Cognate Sets for English and Romance Languages". In: *Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 64–74. DOI: 10.18653/v1/2021.lchange-1.9 (cit. on pp. 16, 148).

[302] A.-S. Uban, A.-M. Ciobanu, and L. P. Dinu. "A Computational Approach to Measuring the Semantic Divergence of Cognates". In: *Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing*. Dec. 2020 (cit. on p. 16).

[303] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett. 2017, pp. 5998–6008 (cit. on pp. 32, 34, 50, 61).

[304] S. Vega. "Language". In: *Solitude Standing* (1987) (cit. on p. 11).

[305] P. Vergés Boncompte and M. R. Costa-jussà. "Multilingual Neural Machine Translation: Case-study for Catalan, Spanish and Portuguese Romance Languages". In: *Proceedings of the Fifth Conference on Machine Translation*. Online: Association for Computational Linguistics, Nov. 2020, pp. 447–450 (cit. on pp. 50, 52).

[306] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2 (cit. on p. 106).

[307] E. Voita and I. Titov. "Information-Theoretic Probing with Minimum Description Length". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 183–196. DOI: 10.18653/v1/2020.emnlp-main.14 (cit. on p. 45).

[308] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh. "Universal Adversarial Triggers for Attacking and Analyzing NLP". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2153–2162. DOI: 10.18653/v1/D19-1221 (cit. on p. 45).

[309] A. Wang, J. Hula, P. Xia, R. Pappagari, R. T. McCoy, R. Patel, N. Kim, I. Tenney, Y. Huang, K. Yu, S. Jin, B. Chen, B. Van Durme, E. Grave, E. Pavlick, and S. R. Bowman. "Can You Tell Me How to Get Past Sesame Street? Sentence-Level Pretraining Beyond Language Modeling". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4465–4476. DOI: 10.18653/v1/P19-1439 (cit. on p. 41).

[310] R. Wang, X. Tan, R. Luo, T. Qin, and T.-Y. Liu. "A Survey on Low-Resource Neural Machine Translation". In: *arXiv:2107.04239 [cs]* (July 2021). arXiv: 2107.04239 (cit. on p. 51).

[311] K. Washio and T. Kato. "Undersampling Improves Hypernymy Prototypicality Learning". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018 (cit. on p. 38).

[312] V. Weber, E. Piovano, and M. Bradford. "It is better to Verify: Semi-Supervised Learning with a human in the loop for large-scale NLU models". In: *Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances*. Online: Association for Computational Linguistics, June 2021, pp. 8–15. DOI: 10.18653/v1/2021.dash-1.2 (cit. on p. 43).

[313] P. Werbos. "Backpropagation through time: what it does and how to do it". In: *Proceedings of the IEEE* 78.10 (Oct. 1990), pp. 1550–1560. DOI: 10.1109/5.58337 (cit. on p. xiii).

[314] S. Wiegreffe and Y. Pinter. "Attention is not not Explanation". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 11–20. DOI: 10.18653/v1/D19-1002 (cit. on p. 46).

[315] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. "The marginal value of adaptive gradient methods in machine learning". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp. 4151–4161 (cit. on p. 42).

[316] J. Wood, T. Andersson, A. Bachem, C. Best, F. Genova, D. R. Lopez, W. Los, M. Marinucci, L. Romary, H. Van de Sompel, J. Vigen, P. Wittenburg, D. Giaretta, and R. L. Hudson. *Riding the Wave. How Europe can gain from the rising tide of scientific data*. Final report of the High Level Expert Group on Scientific Data. European Comission, Oct. 2010, p. 40 (cit. on p. 43).

[317] W. Wu and D. Yarowsky. "Creating Large-Scale Multilingual Cognate Tables". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018 (cit. on pp. 16, 17, 145).

[318] W. Wu and D. Yarowsky. "Computational Etymology and Word Emergence". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 3252–3259 (cit. on p. 18).

[319] M. Xia, X. Kong, A. Anastasopoulos, and G. Neubig. "Generalized Data Augmentation for Low-Resource Translation". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5786–5796. DOI: 10.18653/v1/P19-1579 (cit. on p. 52).

[320] J. Xu, J. Crego, and J. Senellart. "Boosting Neural Machine Translation with Similar Translations". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 1580–1590. DOI: 10.18653/v1/2020.acl-main.144 (cit. on p. 52).

[321] M. Yang, S. Liu, K. Chen, H. Zhang, E. Zhao, and T. Zhao. "A Hierarchical Clustering Approach to Fuzzy Semantic Representation of Rare Words in Neural Machine Translation". In: *IEEE Transactions on Fuzzy Systems* 28.5 (May 2020), pp. 992–1002. DOI: 10.1109/TFUZZ.2020.2969399 (cit. on pp. 42, 53).

[322] Z. Yang, A. Einolghozati, H. Inan, K. Diedrick, A. Fan, P. Donmez, and S. Gupta. "Improving Text-to-Text Pre-trained Models for the Graph-to-Text Task". In: *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*. Dublin, Ireland (Virtual): Association for Computational Linguistics, Dec. 2020, pp. 107–116 (cit. on p. 53).

[323] A. Yazgan and M. Saraclar. "Hybrid language models for out of vocabulary word detection in large vocabulary conversational speech recognition". In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. May 2004, pp. I–745. DOI: 10.1109/ICASSP.2004.1326093 (cit. on p. 42).

[324] E. Zeidler and D. Crévenat-Werner. *Orthographe alsacienne: bien écrire l'alsacien de Wissembourg à Ferrette*. Colmar, France: Jérôme Do Bentzinger, 2008 (cit. on p. 131).

[325] B. Zhang, D. Xiong, J. Su, Q. Liu, R. Ji, H. Duan, and M. Zhang. "Variational Neural Discourse Relation Recognizer". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 382–391. DOI: 10.18653/v1/D16-1037 (cit. on p. 54).

[326] J.-J. Zhang, F.-F. Zhai, and C.-Q. Zong. "A Substitution-Translation-Restoration Framework for Handling Unknown Words in Statistical Machine Translation". In: *Journal of Computer Science and Technology* 28.5 (Sept. 2013), pp. 907–918. DOI: 10.1007/s11390-013-1386-5 (cit. on p. 54).

[327] J. Zhang, F. Zhai, and C. Zong. "Handling Unknown Words in Statistical Machine Translation from a New Perspective". In: *Natural Language Processing and Chinese Computing*. Ed. by M. Zhou, G. Zhou, D. Zhao, Q. Liu, and L. Zou. Communications in Computer and Information Science. Berlin, Heidelberg: Springer, 2012, pp. 176–187. DOI: 10.1007/978-3-642-34456-5_17 (cit. on p. 54).

[328] K. Zhang and S. Bowman. "Language Modeling Teaches You More than Translation Does: Lessons Learned Through Auxiliary Syntactic Task Analysis". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 359–361. DOI: 10.18653/v1/W18-5448 (cit. on p. 123).

[329] W. Zhao, G. Glavaš, M. Peyrard, Y. Gao, R. West, and S. Eger. "On the Limitations of Cross-lingual Encoders as Exposed by Reference-Free Machine Translation Evaluation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 1656–1671. DOI: 10.18653/v1/2020.acl-main.151 (cit. on p. 51).

[330] R. Zhong, D. Ghosh, D. Klein, and J. Steinhardt. "Are Larger Pretrained Language Models Uniformly Better? Comparing Performance at the Instance Level". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 3813–3827. DOI: 10.18653/v1/2021.findings-acl.334 (cit. on p. 41).

[331] P. Zhou, J. Feng, C. Ma, C. Xiong, S. C. H. Hoi, et al. "Towards theoretically understanding why sgd generalizes better than adam in deep learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21285–21296 (cit. on p. 42).

[332] J. Zhu and E. Hovy. "Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 783–790 (cit. on p. 38).

[333] W. Y. Zou, R. Socher, D. Cer, and C. D. Manning. "Bilingual Word Embeddings for Phrase-Based Machine Translation". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1393–1398 (cit. on p. 54).

APPENDIX

A Linguistic information

We provide, for later reference, a table of the phones of interest studied in this manuscript, as well as their unicodes and tipa commands. We first look at pulmonic consonnants, then vowels, as defined in Section 1.1.1. We also provide a structured table of the pulmonic consonants, which allows to better visualise their respective dimensions (manner and place). We then introduce a list of the different language codes used in this thesis, as well as the reference versions for the Wiktionary.

A.1 Phones of interest and their characteristics

A.1.1 Pulmonic consonants

Symbol	Name	Unicode	TIPA
m	Voiced bilabial nasal	u006D	m
mː	Voiced bilabial geminated nasal	u006D	m:
p	Voiceless bilabial stop	u0070	p
pː	Voiceless bilabial geminated stop	u0070	p:
b	Voiced bilabial stop	u0062	b
bː	Voiced bilabial geminated stop	u0062u02D0	b:
ɸ	Voiceless bilabial fricative	u0278	F
β	Voiced bilabial fricative	u03B2	B
ʙ	Voiced bilabial trill	u0299	\textscb
ɱ	Voiced labiodental nasal	u0271	M
bv	Voiced labiodental affricate	None	bv
f	Voiceless labiodental fricative	u0066	f
fː	Voiceless labiodental geminated fricative	u0066	f:
v	Voiced labiodental fricative	u0076	v
vː	Voiced labiodental geminated fricative	u0076	v:
ʋ	Voiced labiodental approximant	u028B	V
n̪	Voiceless alveolar nasal	None	\textsubsquaren
n	Voiced alveolar nasal	u006E	n
nː	Voiced alveolar geminated nasal	u006E	n:
t	Voiceless alveolar stop	u0074	t
tː	Voiceless alveolar geminated stop	u0074u02D0	t:
d	Voiced alveolar stop	u0064	d
dː	Voiced alveolar geminated stop	u0064u02D0	d:
ts	Voiceless alveolar sibilant affricate	u02A6	ts
dz	Voiced alveolar sibilant affricate	u02A3	dz
tʃ	Voiceless postalveolar sibilant affricate	u0074 u0283	tS
dʒ	Voiced post-alveolar sibilant affricate	u0064 u0292	dZ
tθ	Voiceless dental non-sibilant affricate	None	tT
dð	Voiced dental non-sibilant affricate	None	dD

s	Voiceless alveolar sibilant	u0073	s
sː	Voiceless alveolar geminated sibilant	u0073u02D0	s:
z	Voiced alveolar sibilant	u007A	z
zː	Voiced alveolar geminated sibilant	u007Au02D0	z:
ʃ	Voiceless postalveolar fricative	u0283	S
ʒ	Voiced postalveolar fricative	u0292	Z
ʃː	Voiceless postalveolar geminated fricative	u0283u02D0	S:
ʒː	Voiced postalveolar geminated fricative	u0292u02D0	Z:
θ	Voiceless dental fricative	u03B8	T
ð	Voiced dental fricative	u00F0	D
ɹ	Voiced alveolar approximant	u0279	\textturnr
ɾ	Voiced alveolar tap and flap	u027E	R
ɾː	Voiced alveolar geminated tap and flap	u027E	R:
r	Voiced alveolar trill	u0072	r
rː	Voiced alveolar geminated trill	u0072	r:
tɬ	Voiceless alveolar lateral affricate	u0074 u026C	t \textbeltl
dɮ	Voiced alveolar lateral affrciate	u0064 u026E	d \textlyoghlig
ɬ	Voiceless alveolar lateral fricative	u026C	textbeltl
ɮ	Voiced alveolar lateral fricative	u026E	\textlyoghlig
l	Voiced alveolar lateral approximant	u006C	l
ɺ	Voiced alveolar lateral flaps	u027A	\textturnlonglegr
ɳ	Voiced retroflex nasal	u0273	\textrtailn
ʈ	Voiceless retroflex stop	u0288	\textrtailt
ɖ	Voiced retroflex stop	u0256	\textrtaild
ʈʂ	Voiceless retroflex affricate	u0288 u0282	\textrtailt \textrtails
ɖʐ	Voiced retroflex affricate	u0256 u0290	\textrtaild \textrtailz
ʂ	Voiceless retroflex fricative	u0282	\textrtails
ʐ	Voiced retroflex fricative	u0290	\textrtailz
ɻ	Voiced retroflex approximant	u027B	\textturnrrtail
ɽ	Voiced retroflex flap	u027D	\textrtailr
ɭ	Voiced retroflex lateral approximant	u026D	\textrtaill
ɲ	Voiced palatal nasal	u0272	\textrtailn
c	Voiceless palatal stop	u0063	c
ɟ	Voiced palatal stop	u025F	\textbardotlessj
dʑ	Voiced alveolo-palatal affricate	u02A5	d \textctz
tɕ	Voiceless alveolo-palatal affricate	u02A8	tC
cç	Voiceless palatal affricate	u0063 u00E7	c \cc
ɟʝ	Voiced palatal affricate	u025F u029D	\textbardotlessj J
ɕ	Voiceless alveolo-palatal sibilant fricative	u0255	C
ʑ	Voiced alveolo-palatal sibilant fricative	u0291	\textctz
ç	Voiceless palatal fricative	u00E7	\cc
ʝ	Voiced palatal fricative	u029D	J
j	Voiced palatal approximant	u006A	j
ʎ	Voiced palatal lateral approximant	u028E	textlambda
ŋ	Voiced velar nasal	u014B	N
k	Voiceless velar stop	u006B	k
kː	Voiceless velar geminated stop	u006Bu02D0	k:
k^w	Palatized voiceless velar stop	u006B	k \super w
ɡ	Voiced velar stop	u0261	g
ɡː	Voiced velar geminated stop	u0261	g:
$ɡ^w$	Palatized voiced velar stop	u0261	g \super w
w	Voiced labio-velar approximant	u0077	w
kx	Voiceless velar affricate	None	kx
ɡɣ	Voiced velar affricate	None	gG
x	Voiceless velar fricative	u0078	x
ɣ	Voiced velar fricative	u0263	G

ɰ	Voiced velar approximant	u0270	\textturnmrleg
ʟ	Voiced velar lateral approximant	u029F	\textscl
ɴ	Voiced uvular nasal	u0274	\textscn
q	Voiceless uvular stop	u0071	q
ɢ	Voiced uvular stop	u0262	\textscg
qχ	Voiceless uvular affricate	None	qX
χ	Voiceless uvular fricative	u03C7	X
ʁ	Voiced uvular fricative	u0281	K
ʀ	Voiced uvular trill	u0280	\textscr
ʡ	Epiglottal stop	u02A1	\textbarglotstop
ħ	Voiceless pharyngeal fricative	u0127	\textcrh
ʕ	Voiced pharyngeal fricative	u0295	Q
ʢ	Voiced epiglottal stop or fricative	u02A2	\textbarrevglotstop
ʜ	Voiceless epiglottal trill	u029C	\textsch
ʔ	Glottal stop	u0294	P
ʔh	Voiceless glottal affricate	None	Ph
h	Voiceless glottal fricative	u0068	h
ɦ	Voiced glottal fricative	u0266	H

Table A.1: Some pulmonic consonants

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k g	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap		ⱱ		ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Figure A.1: Official IPA pulmonic consonants chart (CC BY-SA 2020 IPA)

A.1.2 Vowels

Symbol	Name	Unicode	TIPA
i	Close front unrounded vowel	u0069	i
y	Close front rounded vowel	u0079	y
e	Close-mid front unrounded vowel	u0065	e
ø	Close-mid front rounded vowel	u00F8	\o
ɛ	Open-mid front unrounded vowel	u025B	E
ɛ̃	Open-mid front unrounded nasal vowel	u025B	\~ E
œ	Open-mid front rounded vowel	u0153	œ
œ̃	Open-mid front rounded nasal vowel	u0153	\~ \oe
æ	Near-open front unrounded vowel	u00E6	\ae
ã	Open front unrounded nasal vowel	u0061	\~ a
a	Open front unrounded vowel	u0061	a
ɶ	Open front rounded vowel	u0276	\OE
ɪ	Near-close front unrounded vowel	u026A	I
ʏ	Near-close front rounded vowel	u028F	Y
ɨ	Close central unrounded vowel	u0268	1
ʉ	Close central rounded vowel	u0289	0
ɘ	Close-mid central unrounded vowel	u0258	\textreve
ɵ	Close-mid central rounded vowel	u0275	8
ə	mid central vowel	u0259	@
ɜ	Open-mid central unrounded vowel	u025C	3
ɞ	Open-mid central rounded vowel	u025E	\textcloserevepsilon
ɐ̃	Nasalised near-open central vowel	u0250	\~ 5
ɐ	Near-open central vowel	u0250	5
ʊ	Near-close back rounded vowel	u028A	U
ʊ̃	Near-close back rounded nasal vowel	u028A	\~ U
ũ	Near-close back rounded nasal vowel	u028A	\~ u
ɯ	Close back unrounded vowel	u026F	W
u	Close back rounded vowel	u0075	u
ɣ	Close-mid back unrounded vowel	u0264	G
o	Close-mid back rounded vowel	u006F	o
õ	Mid near back rounded nasal vowel	u00F5	\~ o
ʌ	Open-mid back unrounded vowel	u028C	2
ɔ	Open-mid back rounded vowel	u0254	O
ɔ̃	Open-mid back rounded nasal vowel	u0254	\~ O
ɑ	Open back unrounded vowel	u0251	A
ɒ	Open back rounded vowel	u0252	\textturnscripta
iː	Long close front unrounded vowel	u0069	i:
yː	Long close front rounded vowel	u0079	y:
eː	Long close-mid front unrounded vowel	u0065	e:
øː	Long close-mid front rounded vowel	u00F8	\o:
ɛː	Long open-mid front unrounded vowel	u025B	E:
œː	Long open-mid front rounded vowel	u0153	\oe:
æː	Long near-open front unrounded vowel	u00E6	\ae:
aː	Long open front unrounded vowel	u0061	a:
ɶː	Long open front rounded vowel	u0276	\OE:
ɪː	Long near-close front unrounded vowel	u026A	I:
ʏː	Long near-close front rounded vowel	u028F	Y:
ɨː	Long close central unrounded vowel	u0268	1:
ʉː	Long close central rounded vowel	u0289	0:
ɘː	Long close-mid central unrounded vowel	u0258	\textreve
ɵː	Long close-mid central rounded vowel	u0275	8:
əː	Long mid central vowel	u0259	@:

ɜː	Long open-mid central unrounded vowel	u025C	3:
ɞː	Long open-mid central rounded vowel	u025E	\textcloserevepsilon :
ɐː	Long near-open central vowel	u0250	5:
ʊː	Long near-close back rounded vowel	u028A	U:
ɯː	Long close back unrounded vowel	u026F	W:
uː	Long close back rounded vowel	u0075	u:
ɣː	Long close-mid back unrounded vowel	u0264	G:
oː	Long close-mid back rounded vowel	u006F	o:
ʌː	Long open-mid back unrounded vowel	u028C	2:
ɔː	Long open-mid back rounded vowel	u0254	O:
ɑː	Long open back unrounded vowel	u0251	A:
ɒː	Long open back rounded vowel	u0252	\textturnscripta:

Table A.2: Some vowels

A.2 Wiktionary language codes

Family	Language	Wiktionary code	Dissertation code
Romance	Aromanian	rup	RUP
	Catalan	ca	CA
	French	fr	FR
	Galician	gl	GL
	Italian	it	IT
	Latin – Classical	la	LA
	Latin – Vulgar	lat-vul	LA
	Middle French	frm	
	Occitan	oc	OC
	Old Catalan	roa-oca	
	Old French	fro	
	Old Latin	itc-ola	
	Old Occitan	pro	
	Old Portuguese	roa-opt	
	Old Spanish	osp	
	Portuguese	pt	PT
	Proto-Indo European	ine-pro	
	Proto-Italic	itc-pro	
	Romanian	ro	RO
	Spanish	es	ES
Germanic	Alsatian	gsw	ALS
	German	de	DE
	Germanic Swiss	gsw	GSW
	Luxembourgish	lb	LB

Table A.3: Wiktionary language code of our languages of interest in our chosen database, versus languages codes we used in the thesis.
The wiktionary code for Alsatian is gsw, because it does not distinguish between Alemannic languages, all under the 'gsw tag umbrella'.

B Artificial data experiment results

In this section, we display the detailed tables of all the experiments we realised when looking for the best hyperparameters combinations using artificially generated data, in Section 6.2. We successively optimized parameters by group, selecting the best combination at one step as an initialisation for the following steps of the research. (The artificial data followed phonotactics and phonetic rules of the Romance language family).

B.1 Hyperparameter search for bilingual data

B.1.1 Embedding dimension versus hidden size

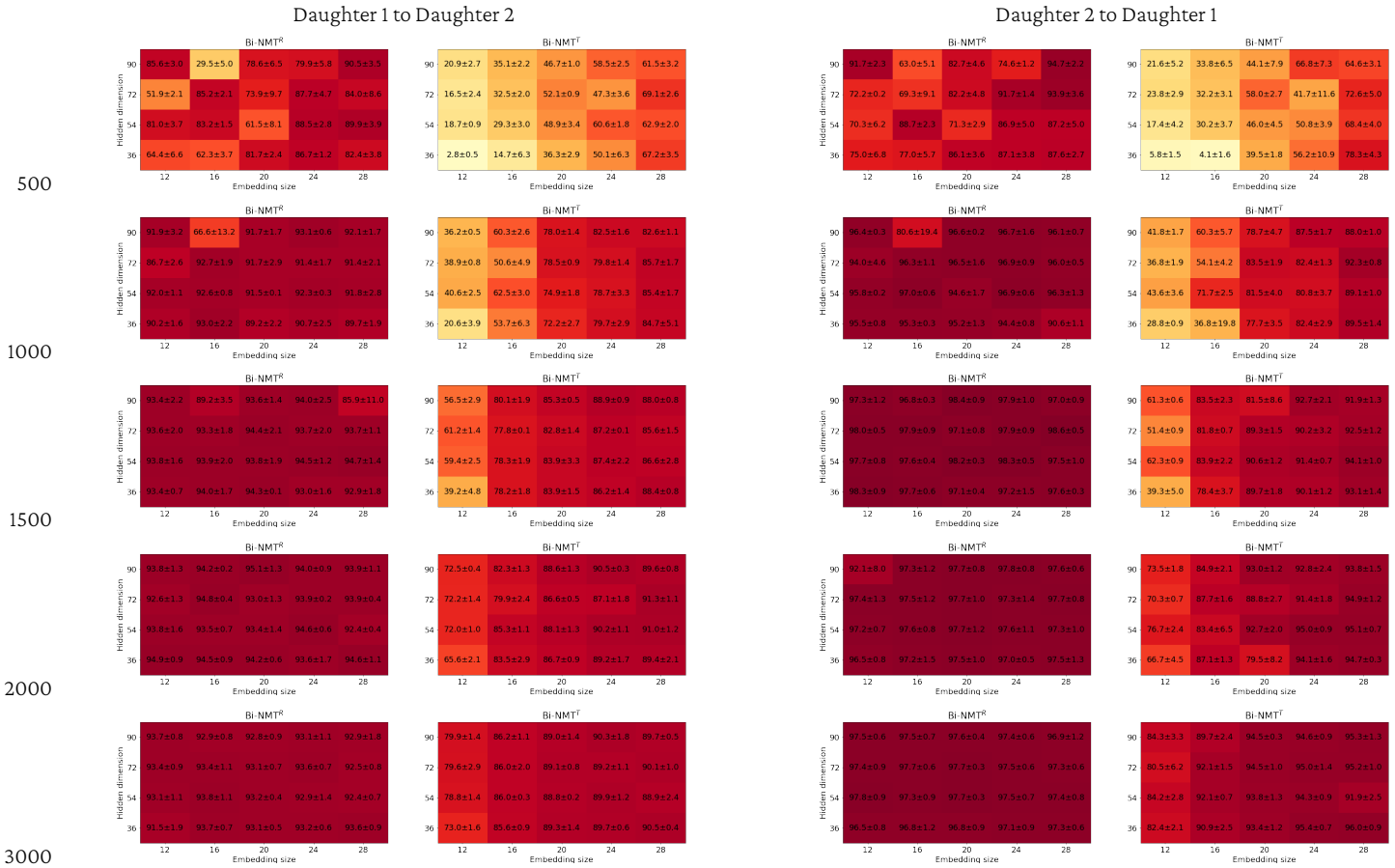


Figure B.1: BLEU results of our experiments on artificial data.
Comparing embedding dimension versus hidden size, for our five data sizes.
From top to bottom: 500, 1000, 1500, 2000, 3000.

Figure B.2: BLEU results of our experiments on artificial data.
Comparing embedding dimension versus hidden size, for our five data sizes.
From top to bottom: 500, 1000, 1500, 2000, 3000.

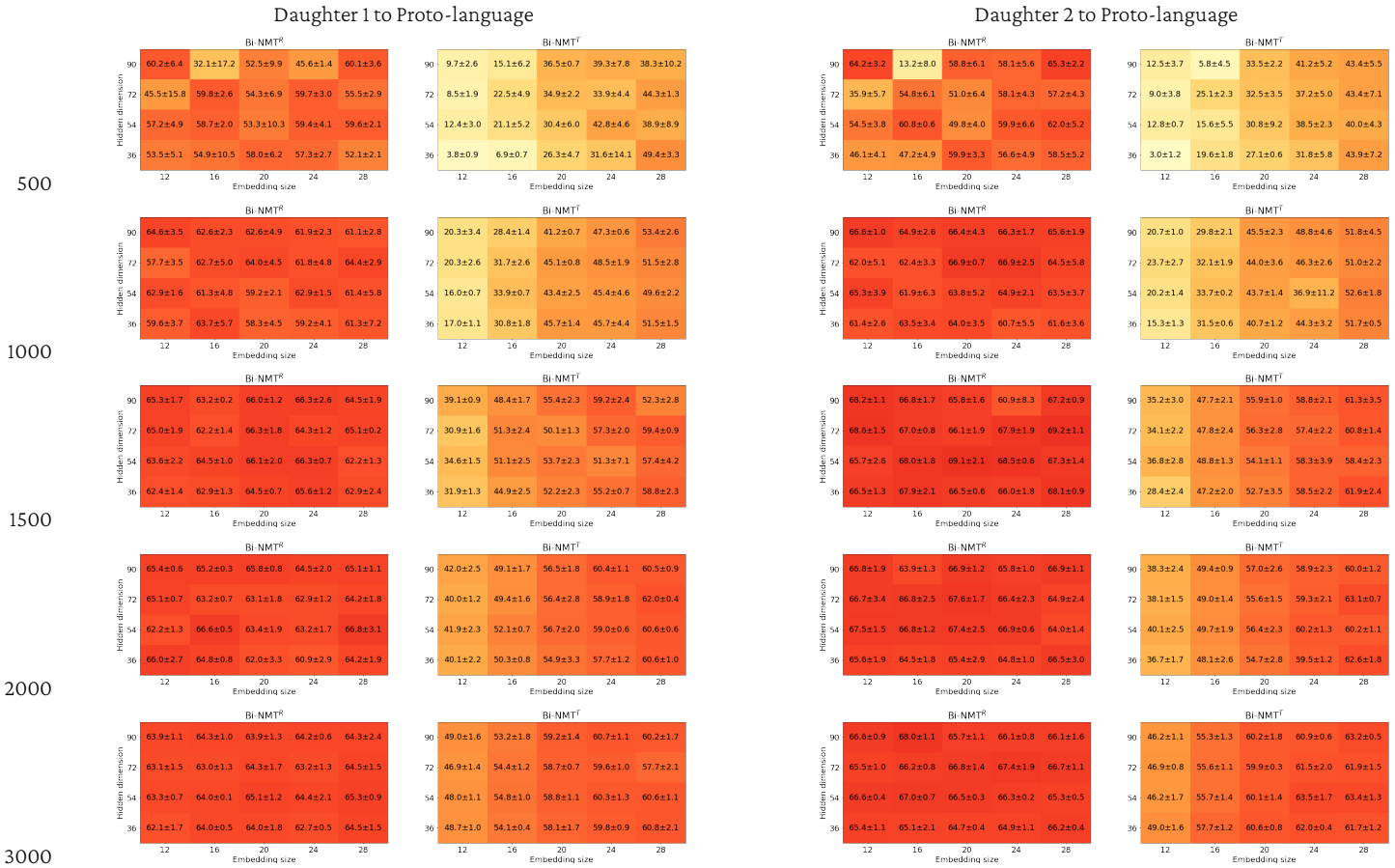


Figure B.3: BLEU results of our experiments on artificial data.
Comparing embedding dimension versus hidden size, for our five data sizes.
From top to bottom: 500, 1000, 1500, 2000, 3000.

B.1.2 Batch size versus learning rate

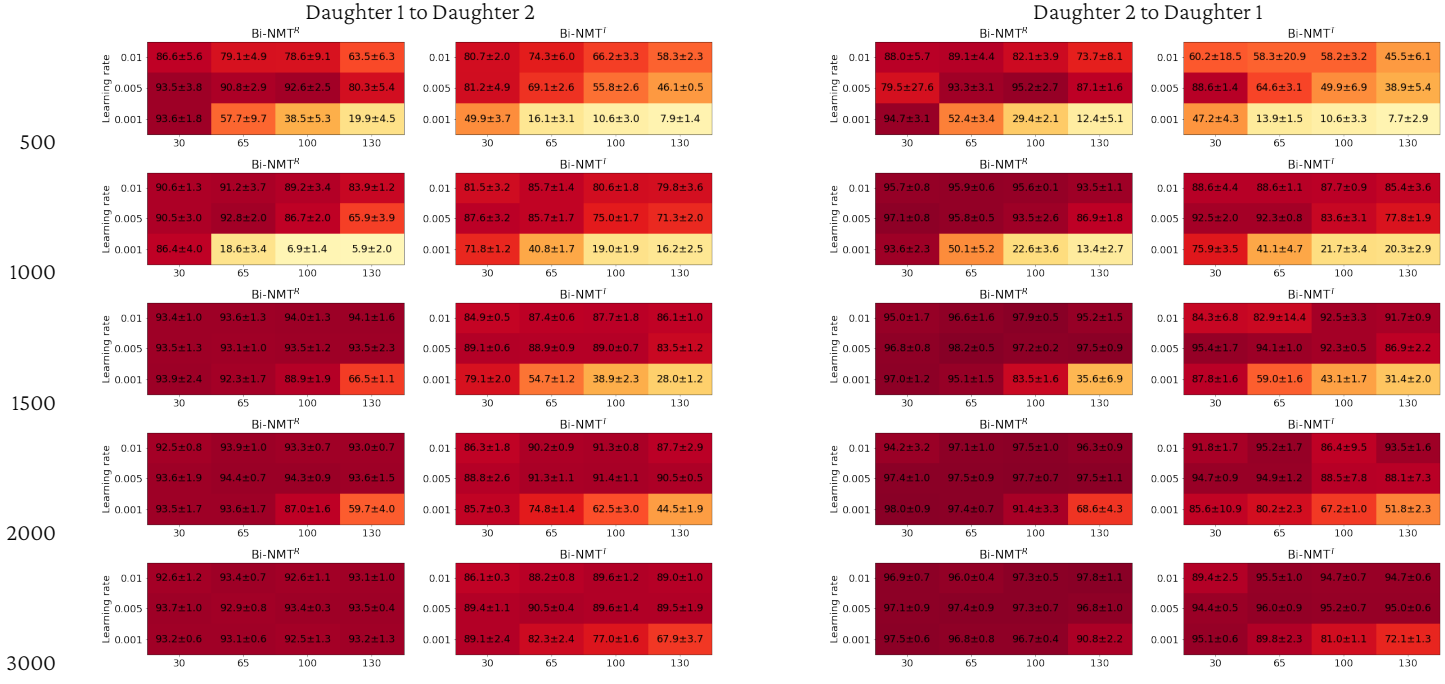


Figure B.4: BLEU results of our experiments on artificial data.
Comparing batch size versus learning rate, for our five data sizes.
From top to bottom: 500, 1000, 1500, 2000, 3000.

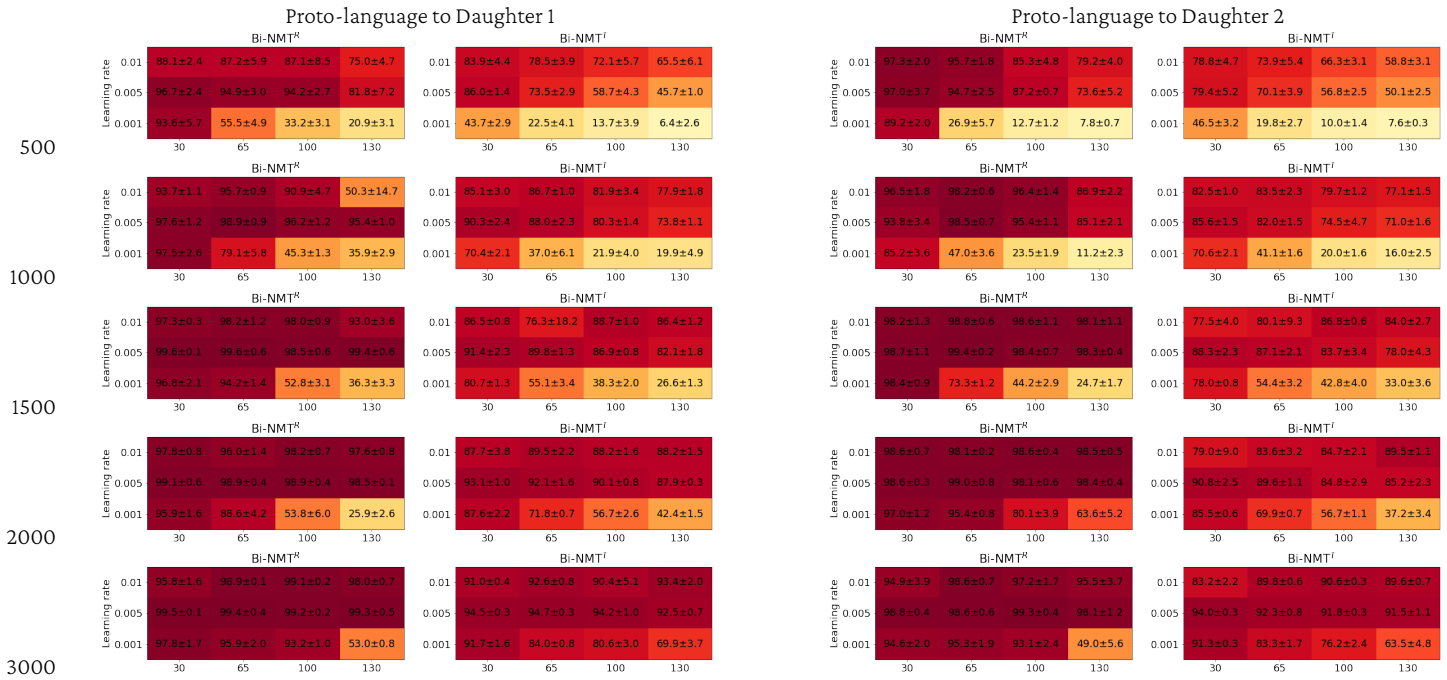


Figure B.5: BLEU results of our experiments on artificial data.
Comparing batch size versus learning rate, for our five data sizes.
From top to bottom: 500, 1000, 1500, 2000, 3000.

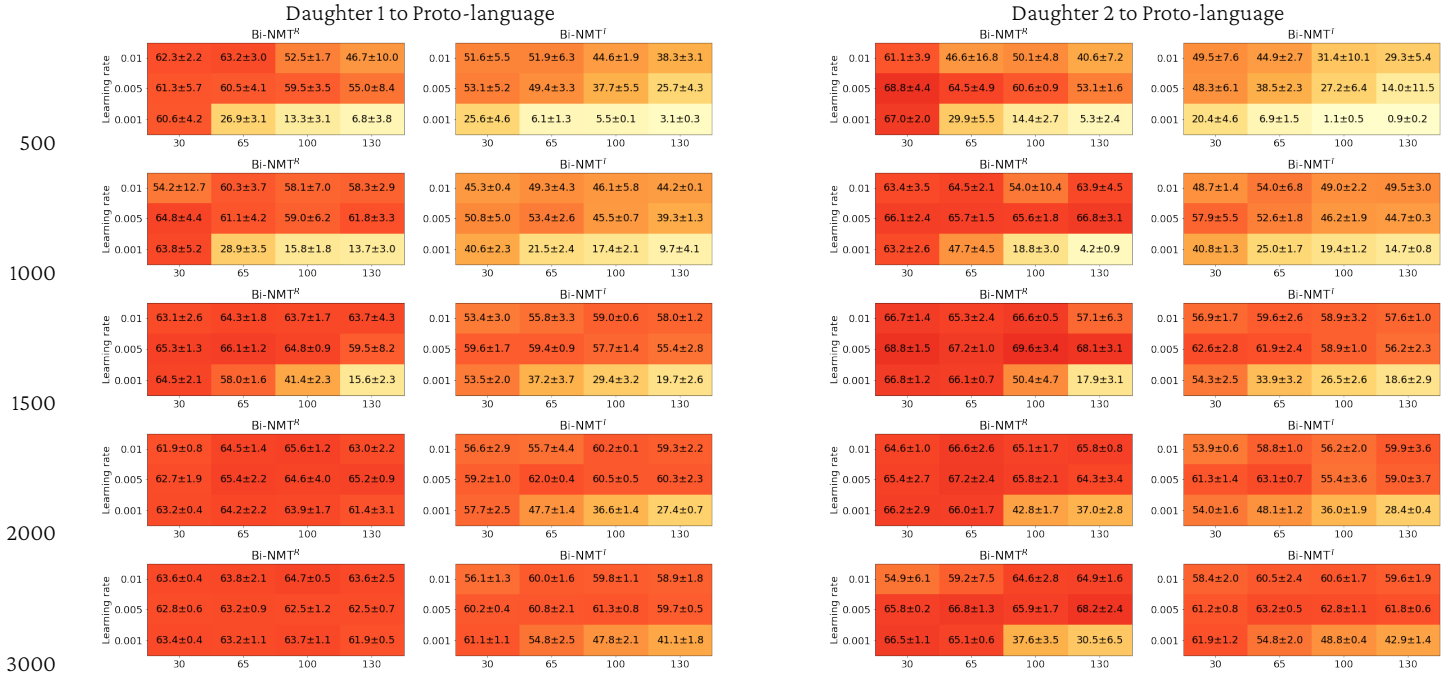


Figure B.6: BLEU results of our experiments on artificial data.
Comparing batch size versus learning rate, for our five data sizes.
From top to bottom: 500, 1000, 1500, 2000, 3000.

B.1.3 Number of layers

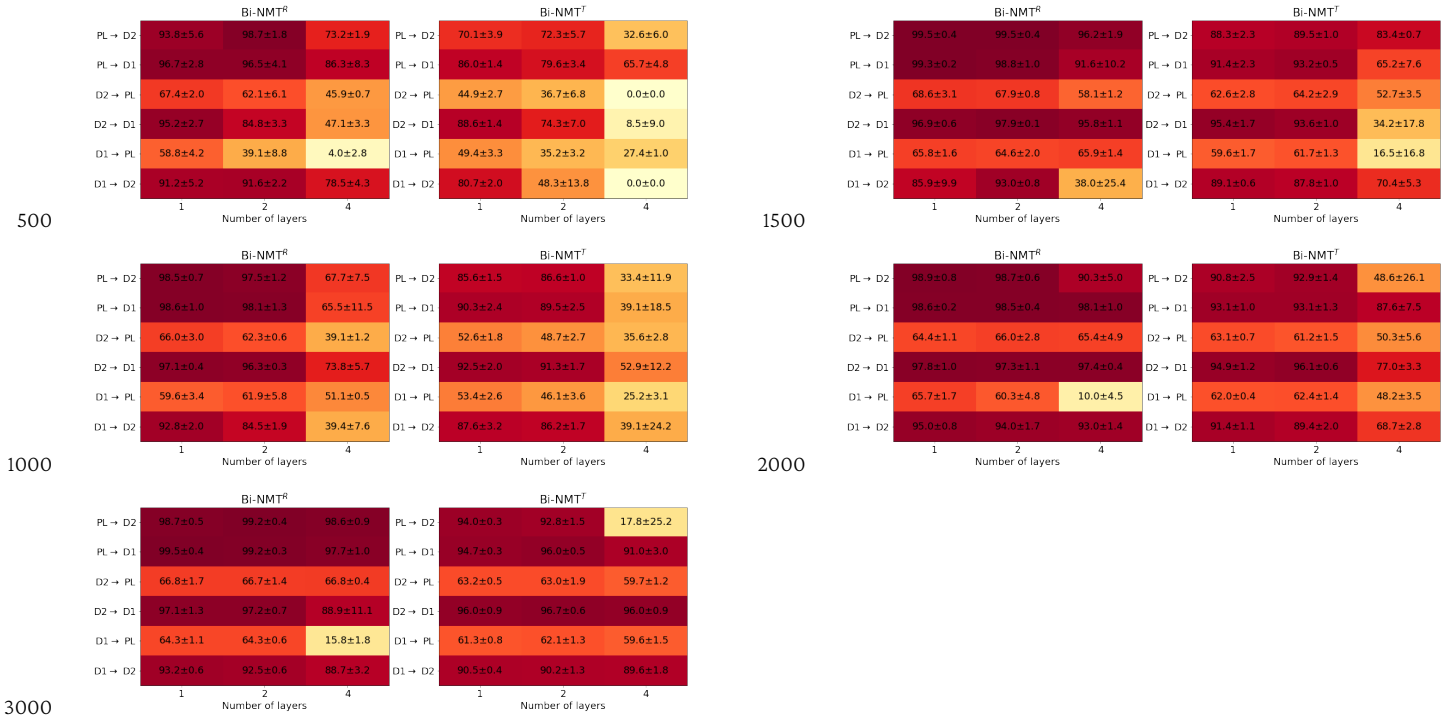


Figure B.7: BLEU results of our experiments on artificial data.
Comparing the impact of the number of layers, for all languages, for our five data sizes.

B.1.4 Transformers: Number of heads

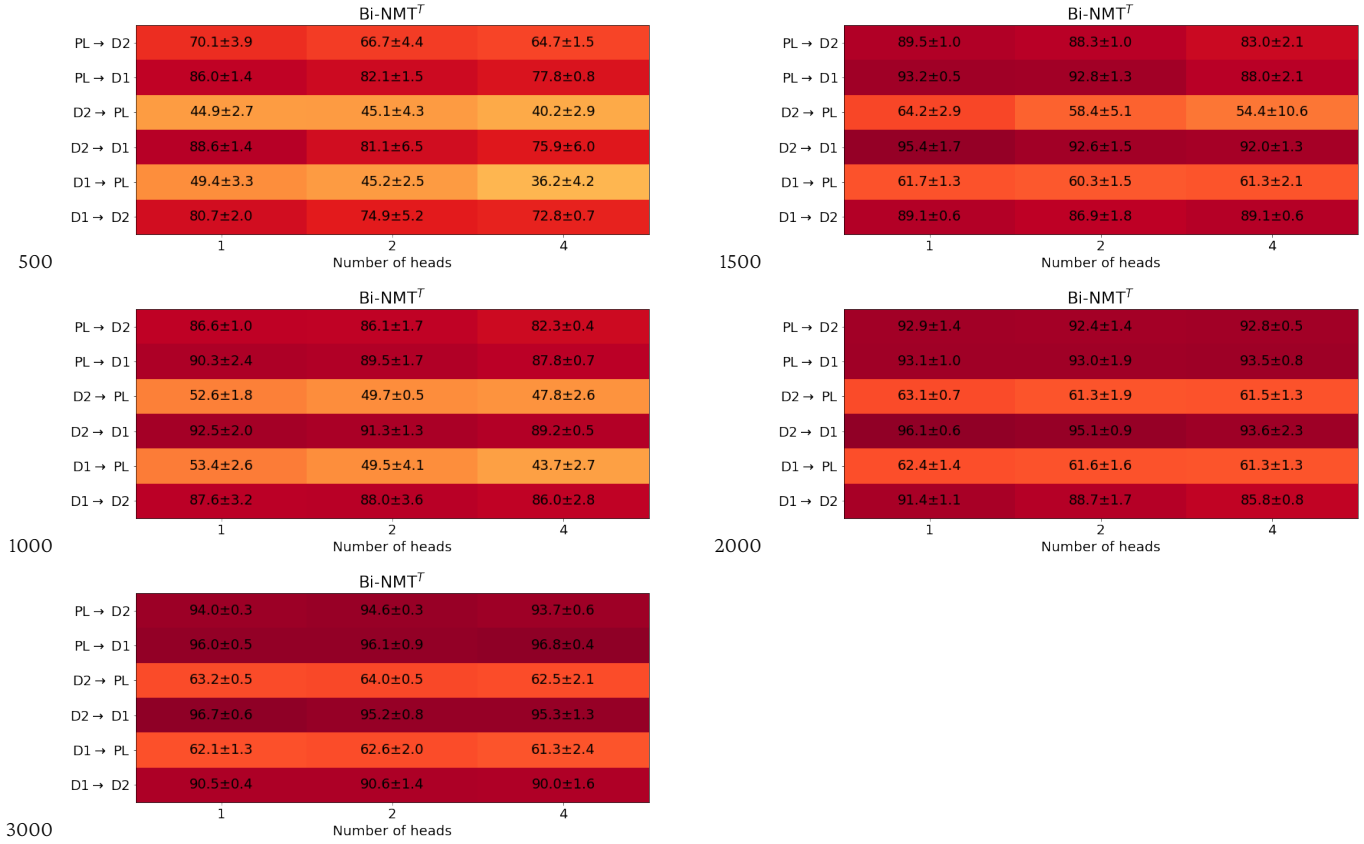


Figure B.8: BLEU results of our experiments on artificial data.
Comparing the impact of the numbe of heads, for all languages, for our five data sizes.

B.1.5 Recurrent models: Attention type

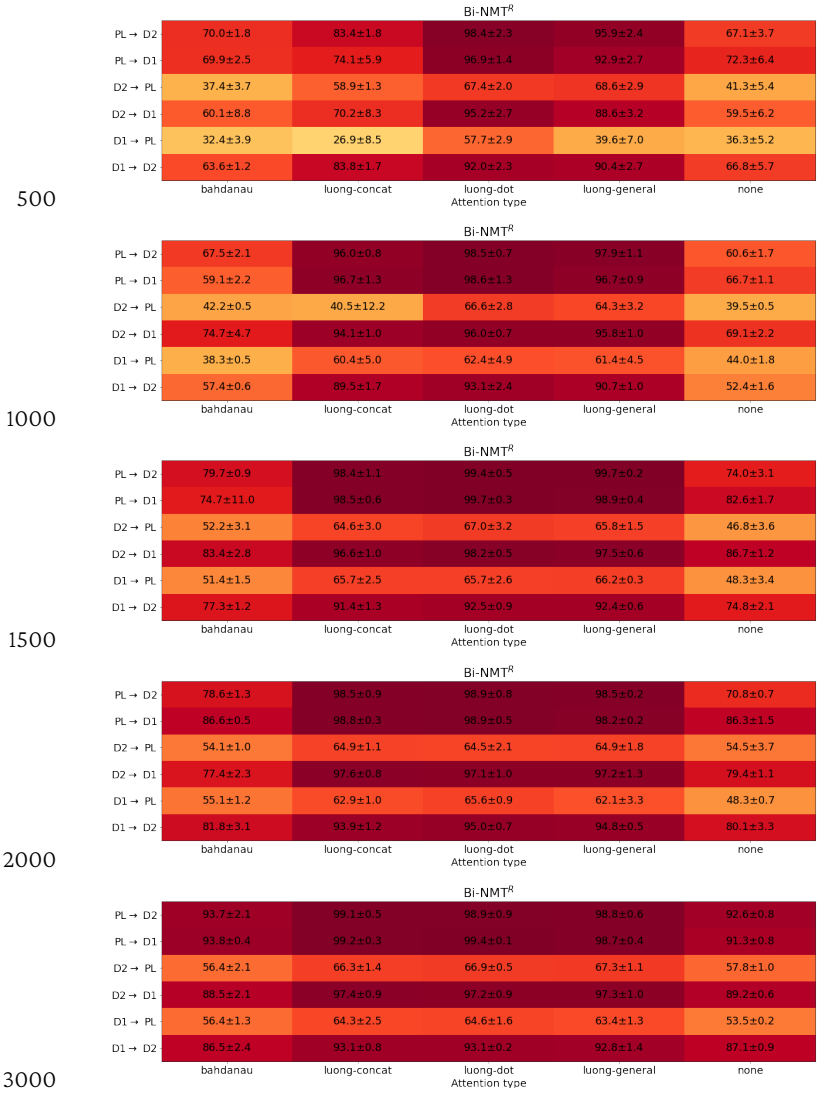


Figure B.9: BLEU results of our experiments on artificial data.
Comparing the impact of the attention type, for all languages, for our five data sizes.

B.2 Hyperparameter search for multilingual data

B.2.1 Embedding dimension versus hidden size

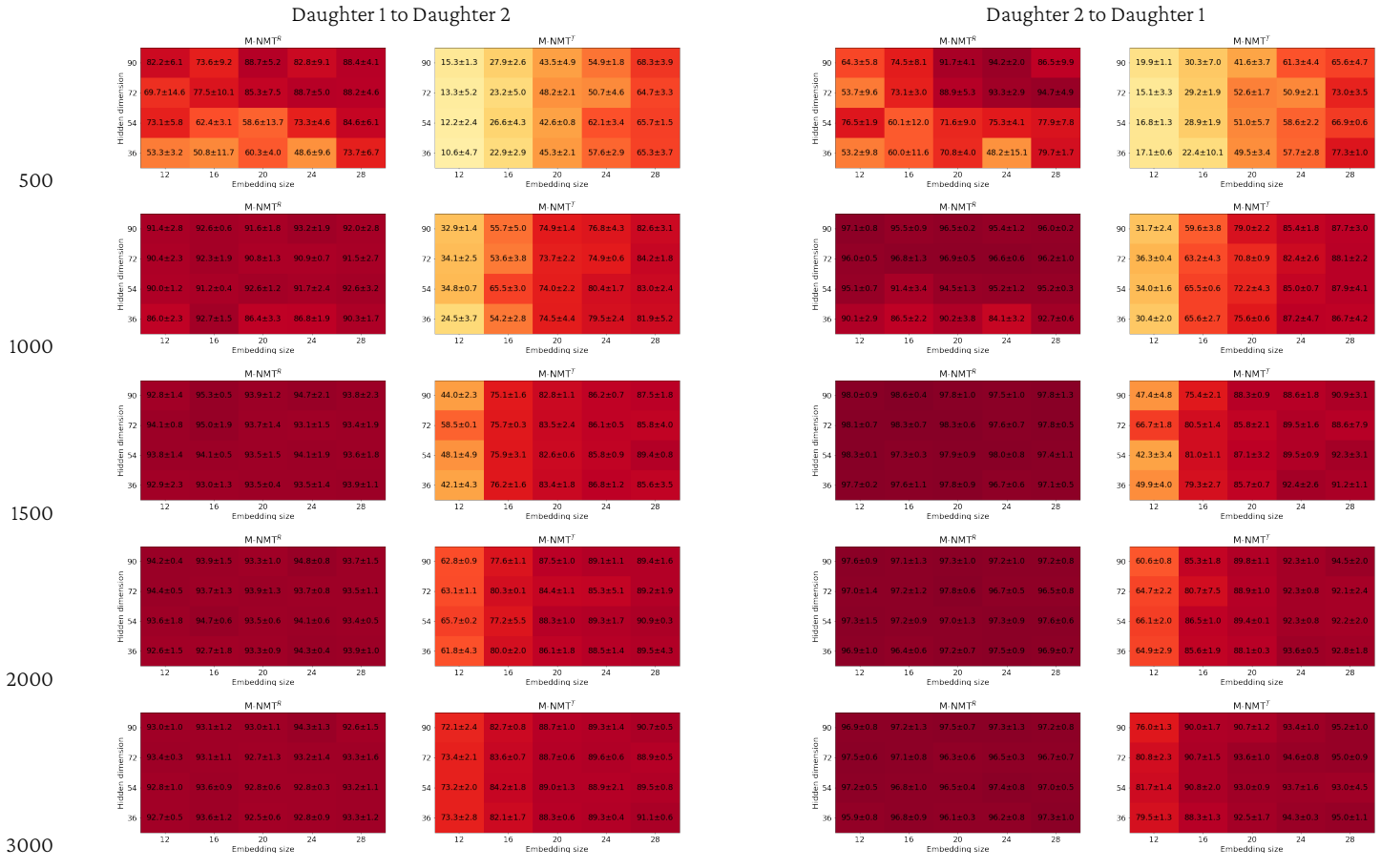


Figure B.10: BLEU results of our experiments on artificial data.
Comparing embedding dimension versus hidden size, for our five data sizes.
From top to bottom: 500, 1000, 1500, 2000, 3000.

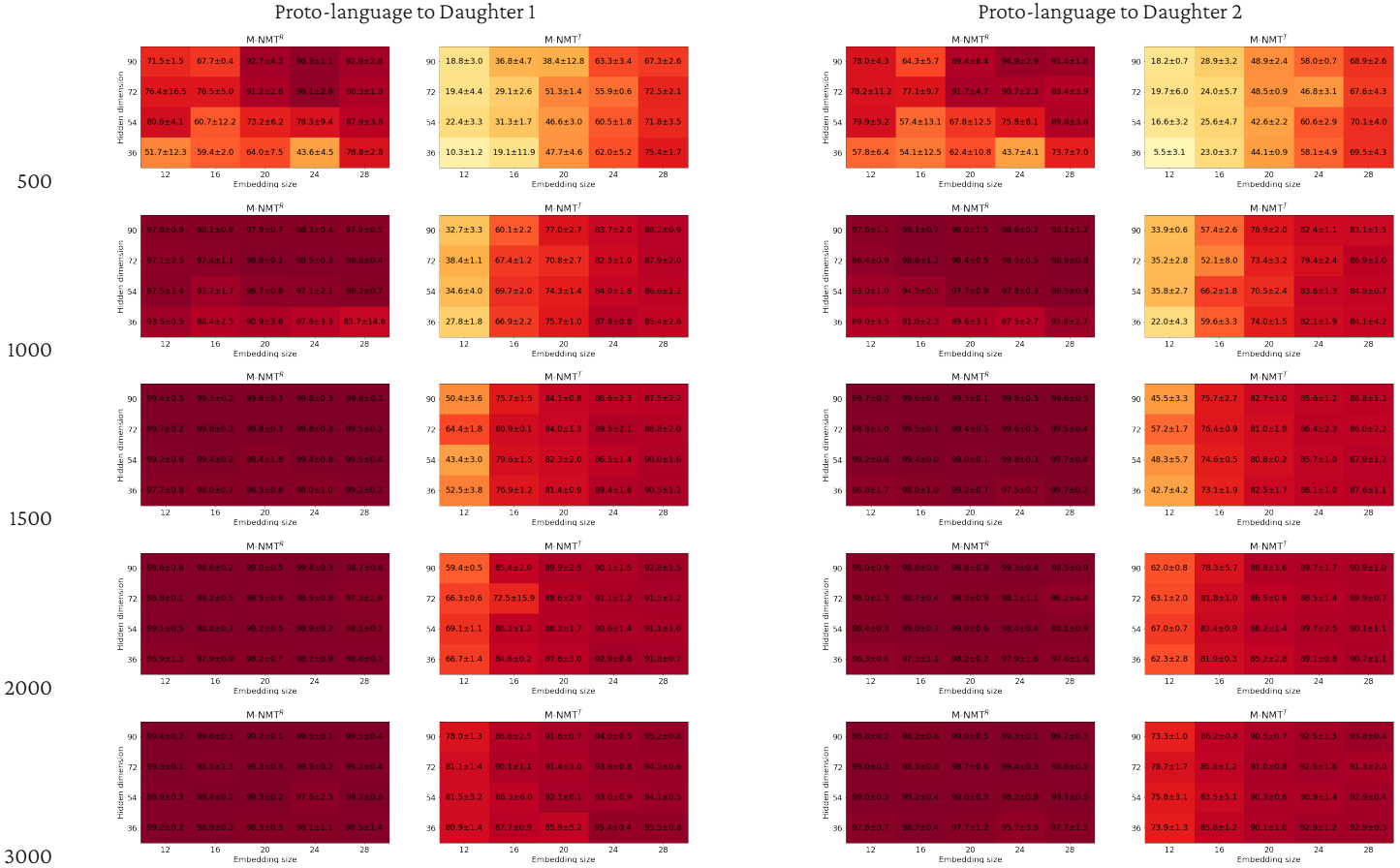


Figure B.11: BLEU results of our experiments on artificial data.
Comparing embedding dimension versus hidden size, for our five data sizes.
From top to bottom: 500, 1000, 1500, 2000, 3000.

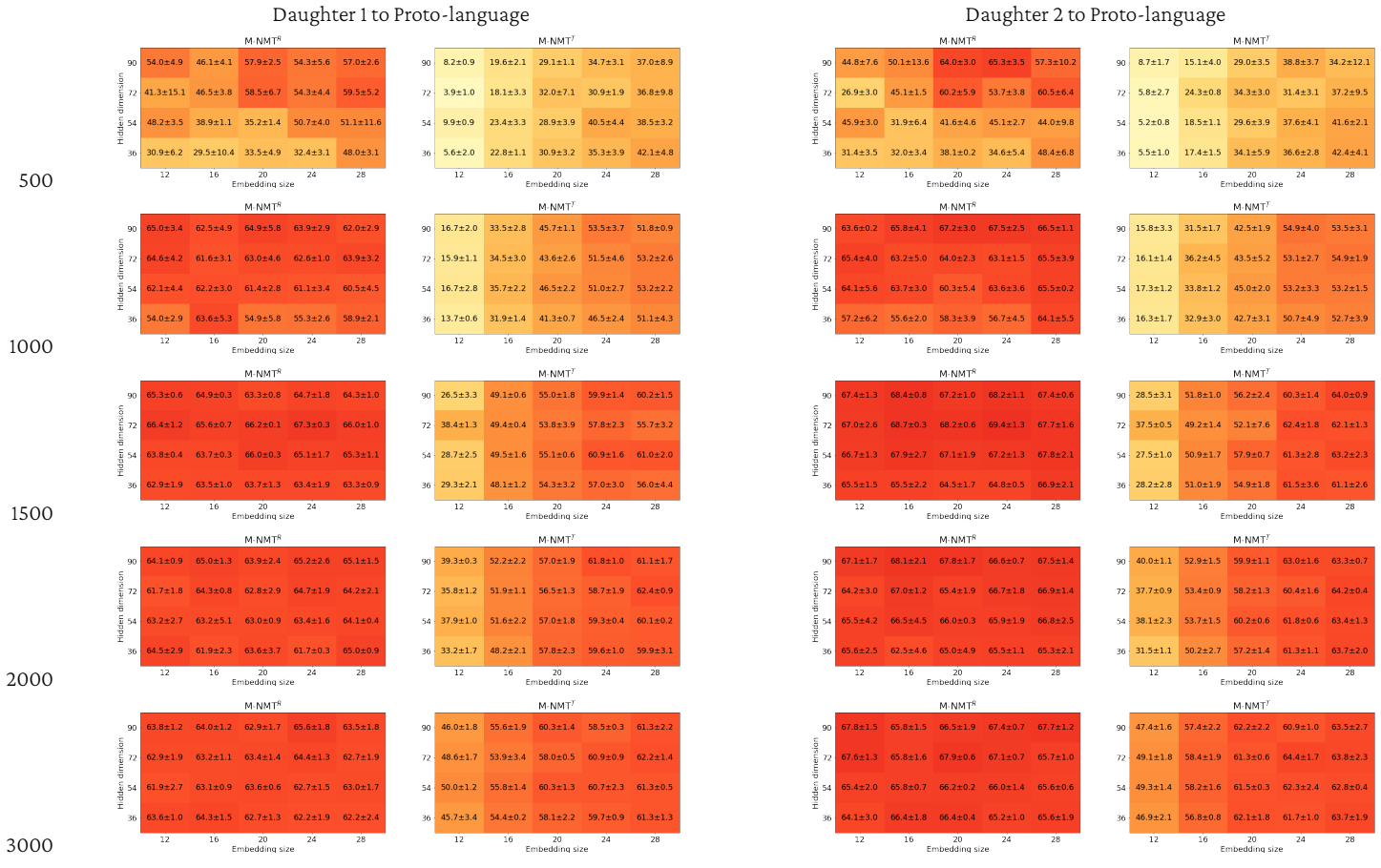


Figure B.12: BLEU results of our experiments on artificial data.
Comparing embedding dimension versus hidden size, for our five data sizes.
From top to bottom: 500, 1000, 1500, 2000, 3000.

B.2.2 Batch size versus learning rate

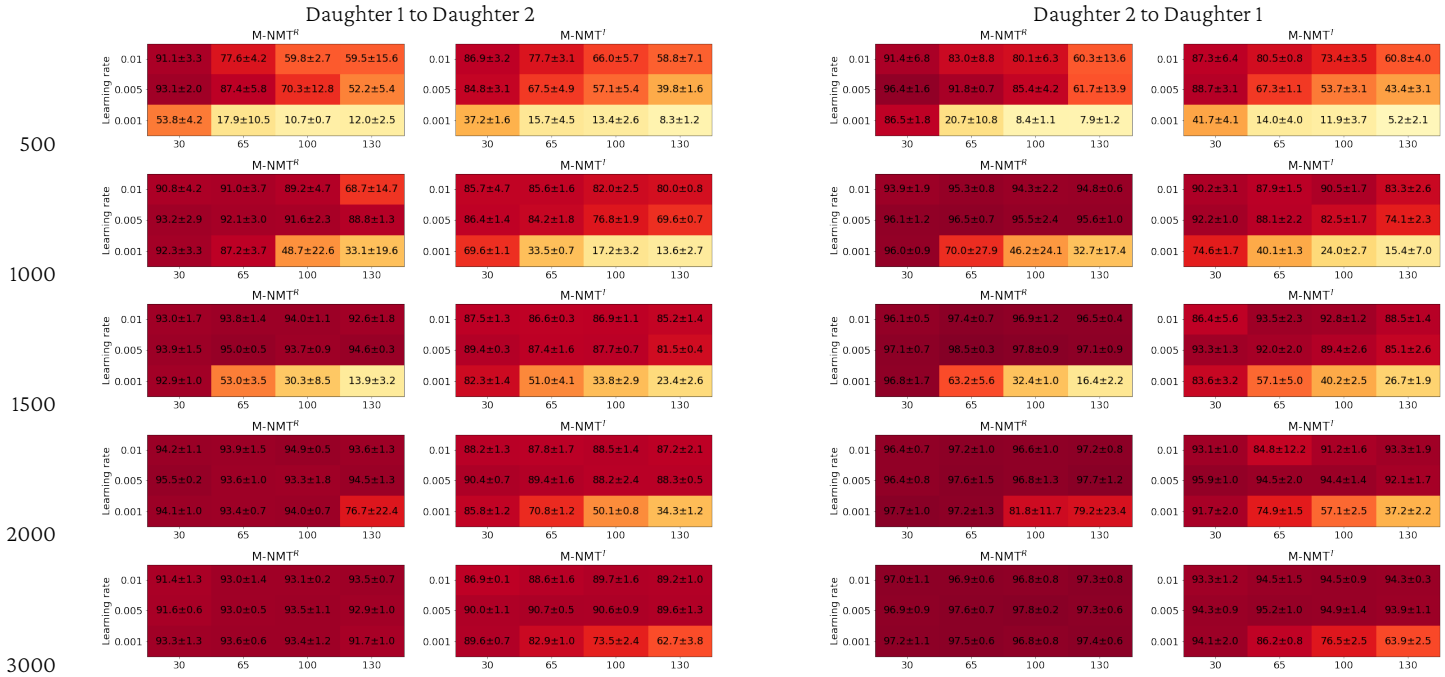


Figure B.13: BLEU results of our experiments on artificial data.
Comparing batch size versus learning rate, for our five data sizes.
From top to bottom: 500, 1000, 1500, 2000, 3000.

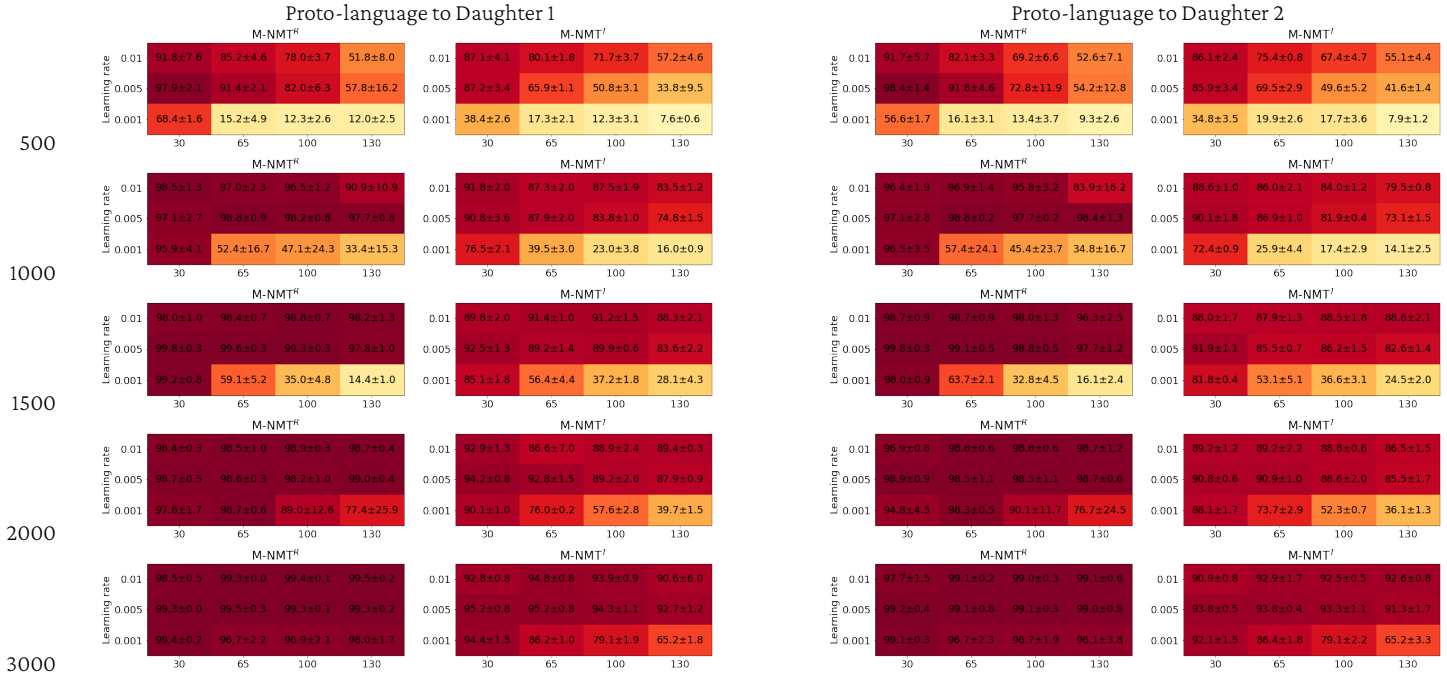


Figure B.14: BLEU results of our experiments on artificial data.
Comparing batch size versus learning rate, for our five data sizes.
From top to bottom: 500, 1000, 1500, 2000, 3000.

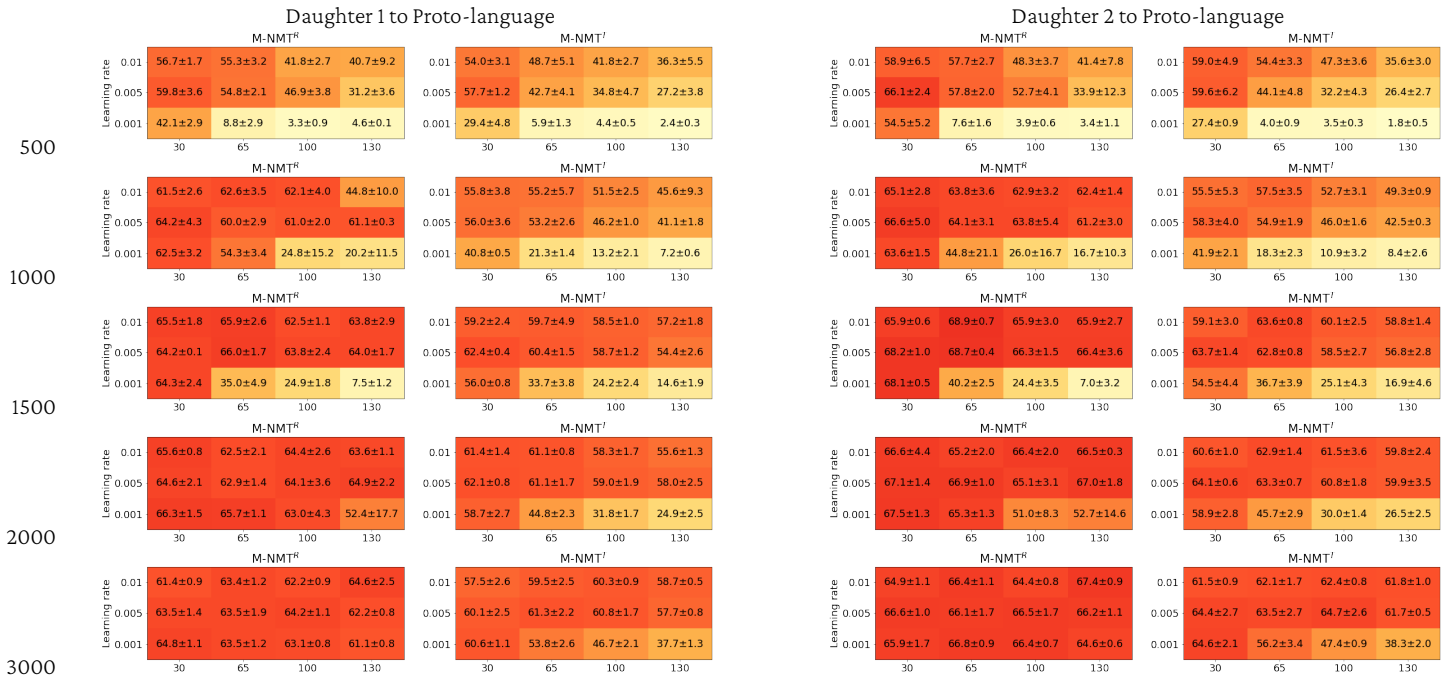


Figure B.15: BLEU results of our experiments on artificial data.
Comparing batch size versus learning rate, for our five data sizes.
From top to bottom: 500, 1000, 1500, 2000, 3000.

B.2.3 Number of layers

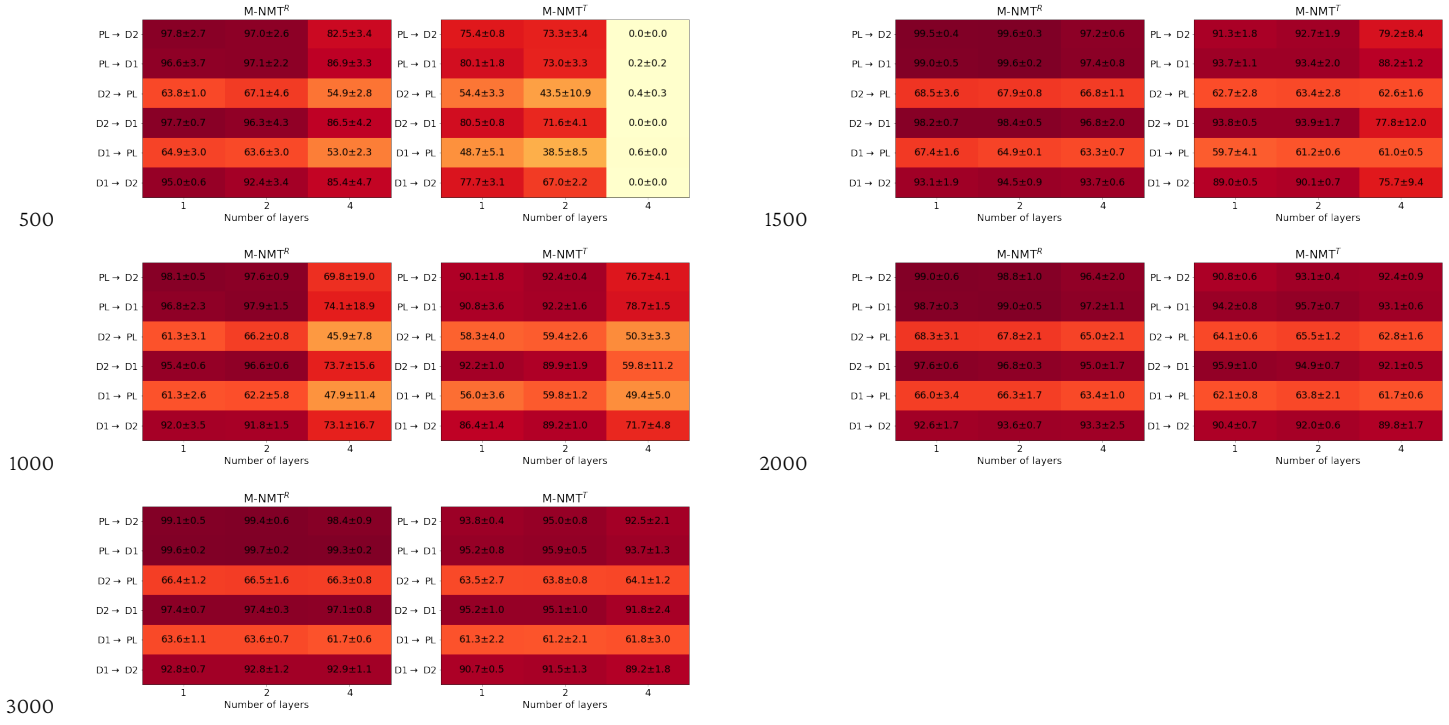


Figure B.16: BLEU results of our experiments on artificial data.
Comparing the impact of the number of layers, for all languages, for our five data sizes.

B.2.4 Transformers: Number of heads

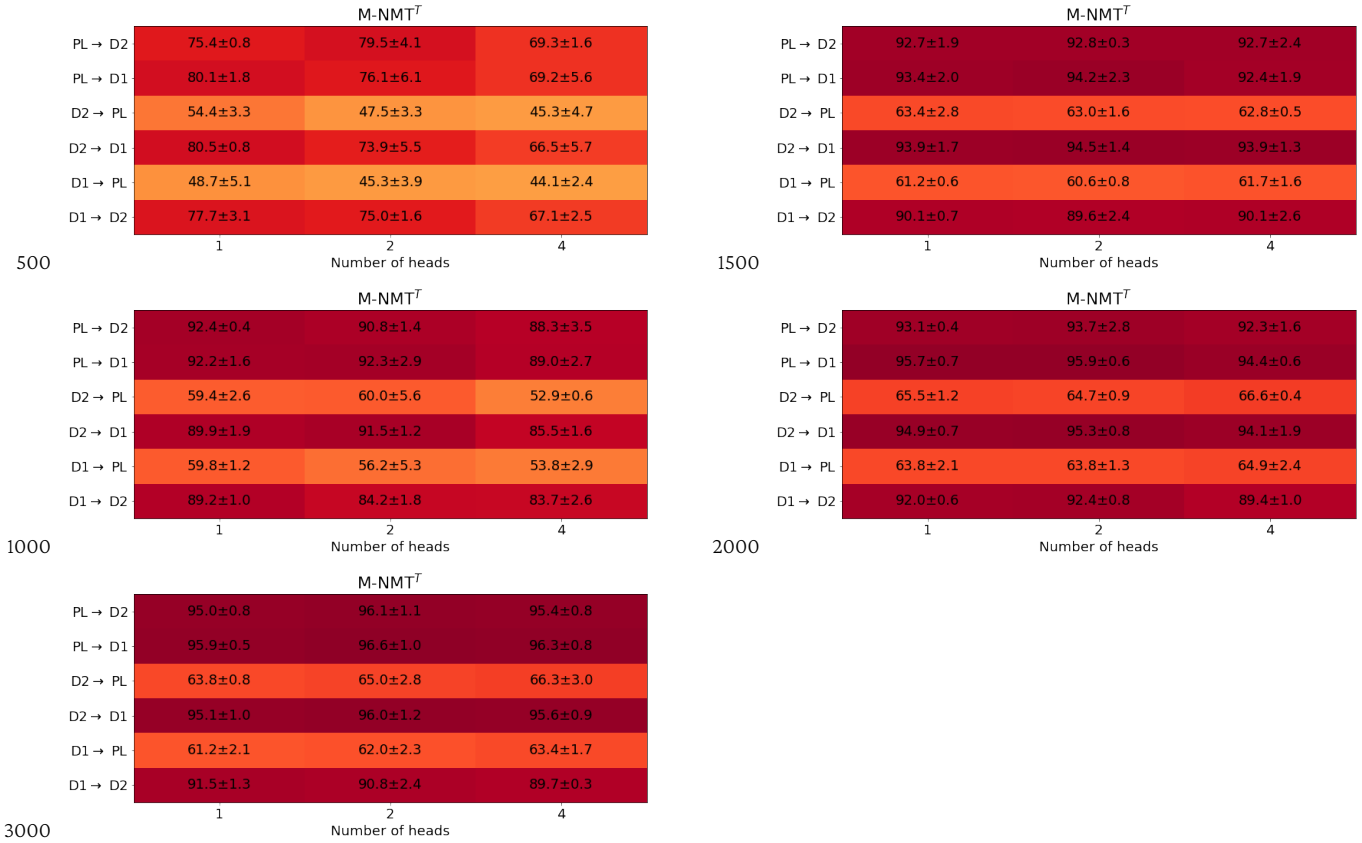


Figure B.17: BLEU results of our experiments on artificial data.
Comparing the impact of the numbe of heads, for all languages, for our five data sizes.

B.2.5 Recurrent models: Attention type

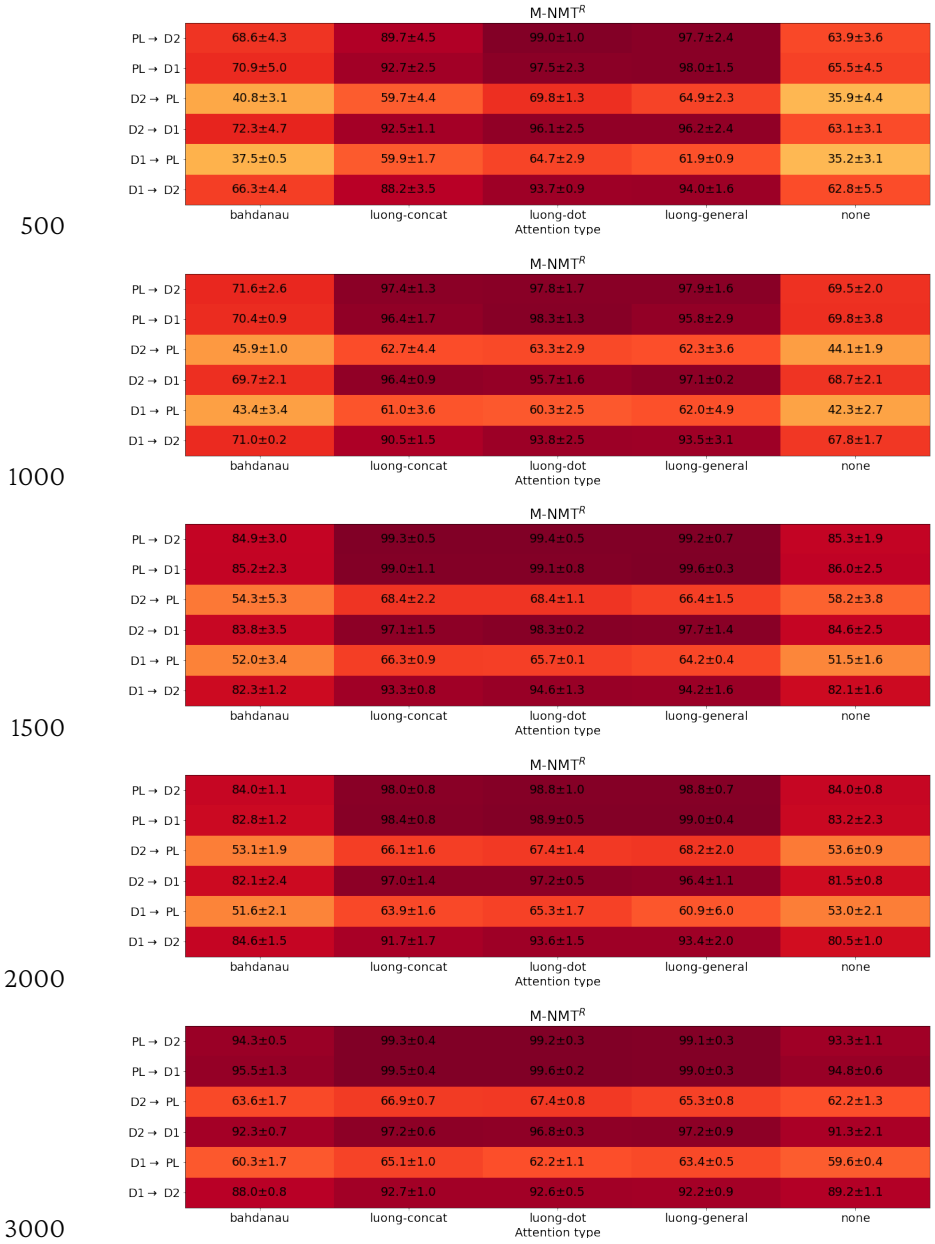


Figure B.18: BLEU results of our experiments on artificial data.
Comparing the impact of the attention type, for all languages, for our five data sizes.

C Real data results

In this section, we display the detailed tables of all the experiments we realised when looking for the best hyperparameters combinations using real Romance data, in Section 6.2. We successively optimized parameters by group, selecting the best combination at one step as an initialisation for the following steps of the research.

C.1 Hyperparameter search for bilingual data

C.1.1 Embedding dimension versus hidden size

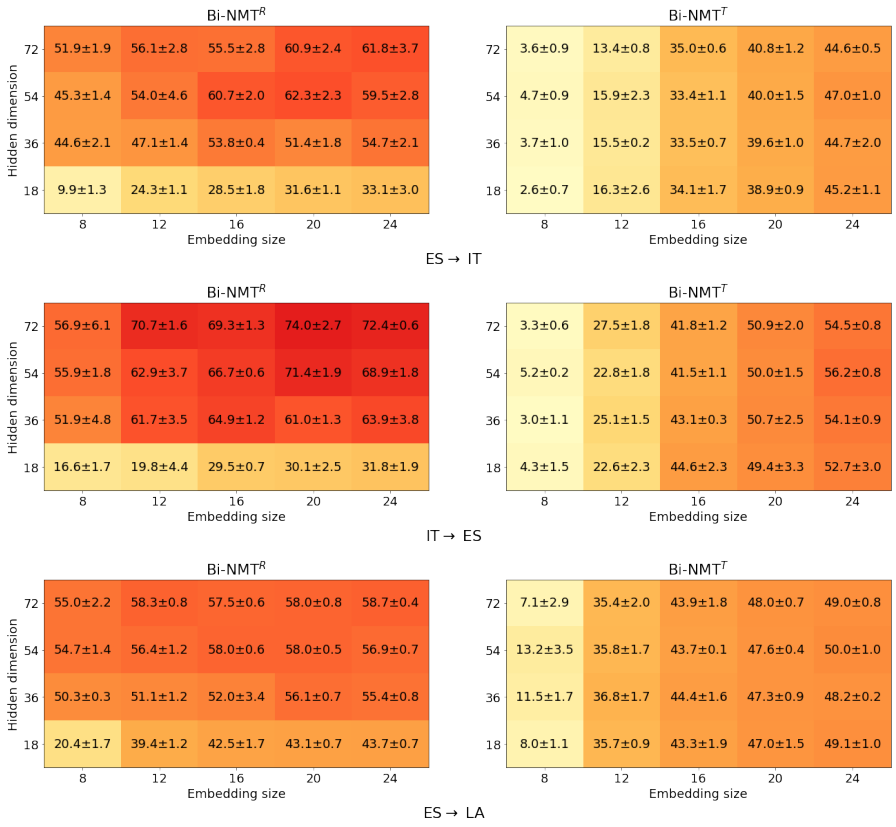


Figure C.1: BLEU results of our experiments on real data - embedding vs hidden size - 1.

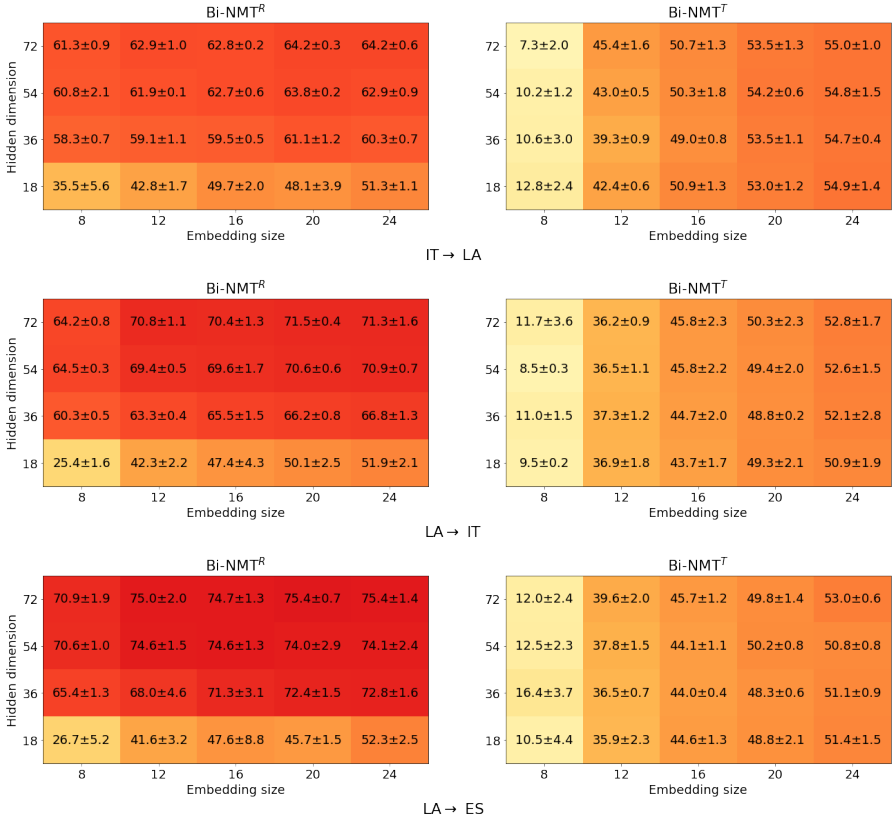


Figure C.2: BLEU results of our experiments on real data - embedding vs hidden size - 2.

C.1.2 Batch size versus learning rate

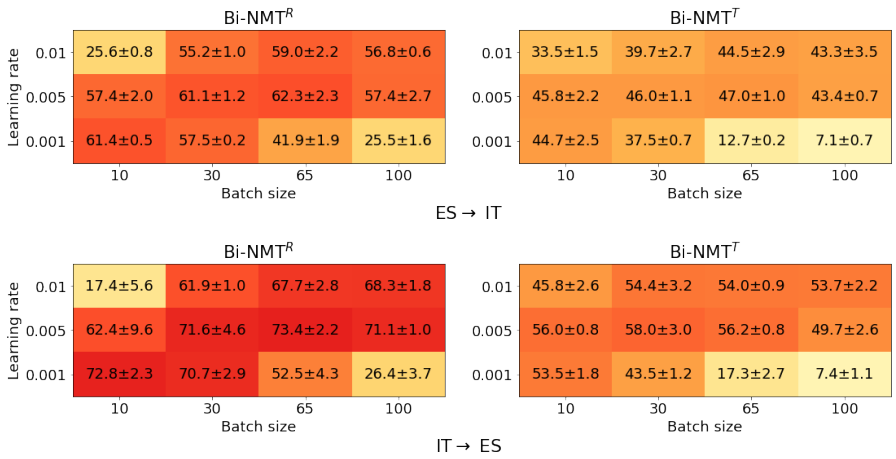


Figure C.3: BLEU results of our experiments on real data - batch size vs learning rate - 1.

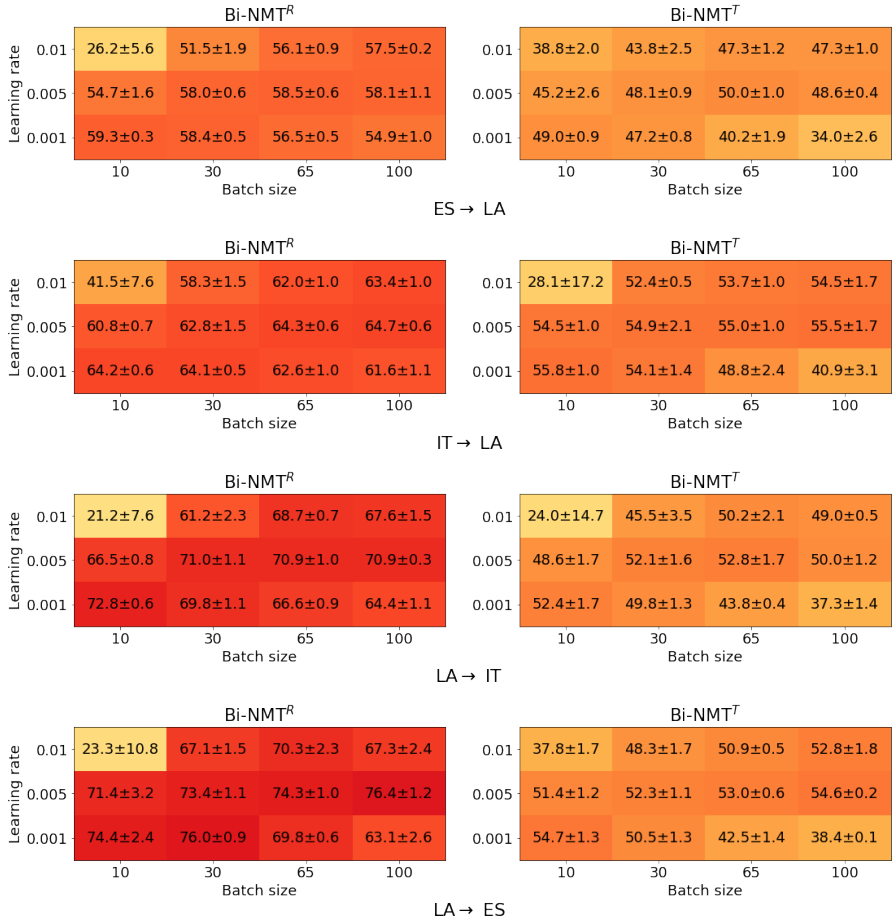


Figure C.4: BLEU results of our experiments on real data - batch size vs learning rate - 2.

C.1.3 Number of layers

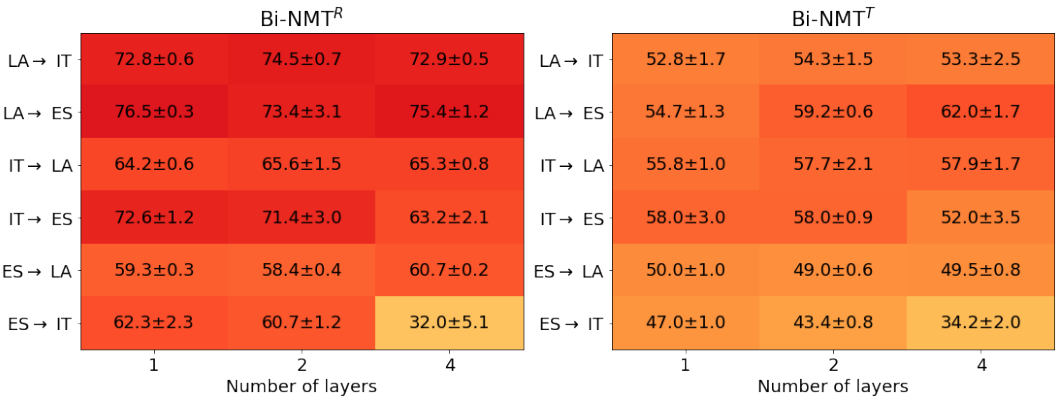


Figure C.5: BLEU results of our experiments on real data - Number of layers

C.1.4 Recurrent models: Attention type

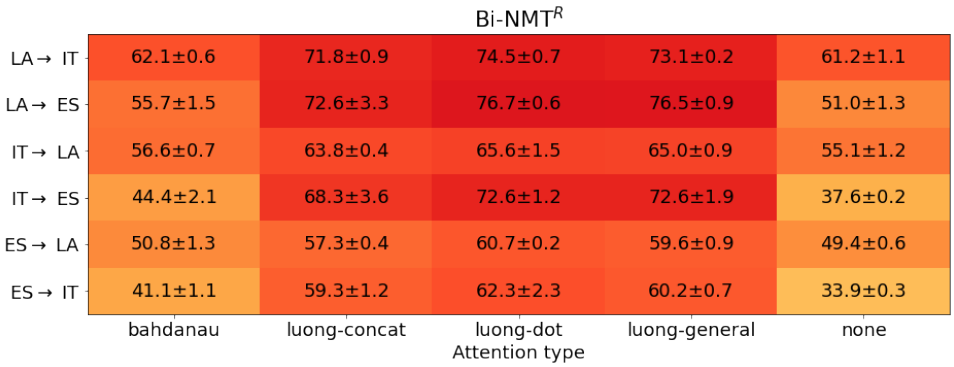


Figure C.6: BLEU results of our experiments on real data - Attention type

C.1.5 Transformers: Number of heads

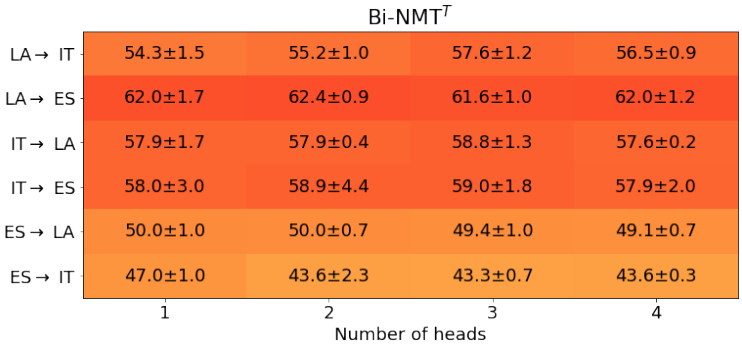


Figure C.7: BLEU results of our experiments on real data - Number of heads

C.2 Hyperparameter search for multilingual data

C.2.1 Embedding dimension versus hidden size

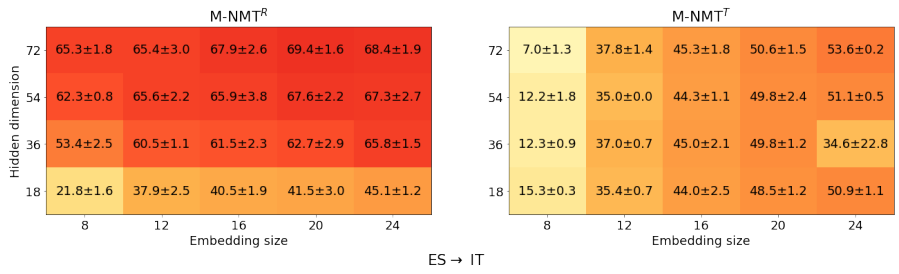


Figure C.8: BLEU results of our experiments on real data - embedding vs hidden size - 1.

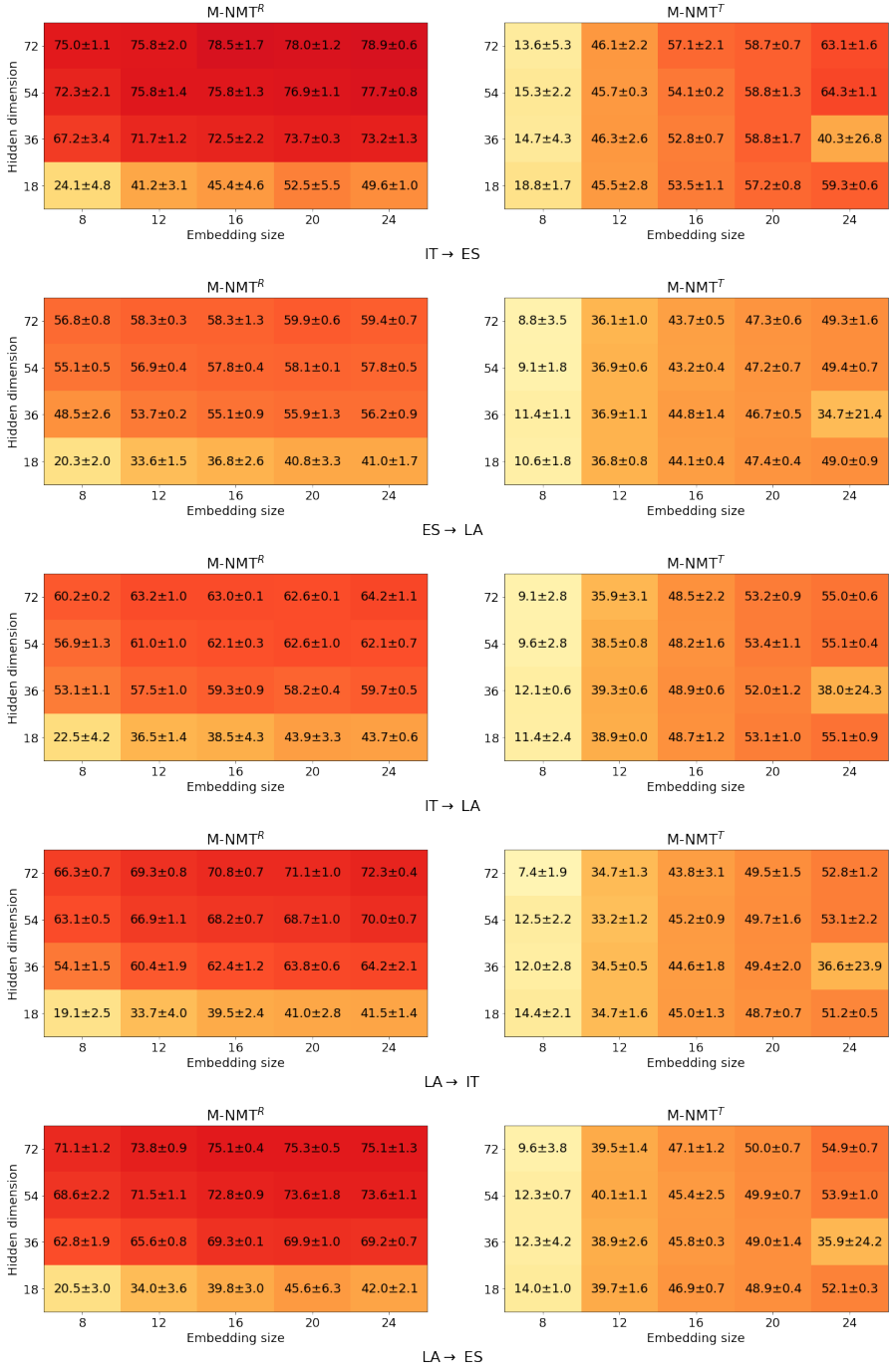


Figure C.9: BLEU results of our experiments on real data - embedding vs hidden size - 2.

C.2.2 Batch size versus learning rate

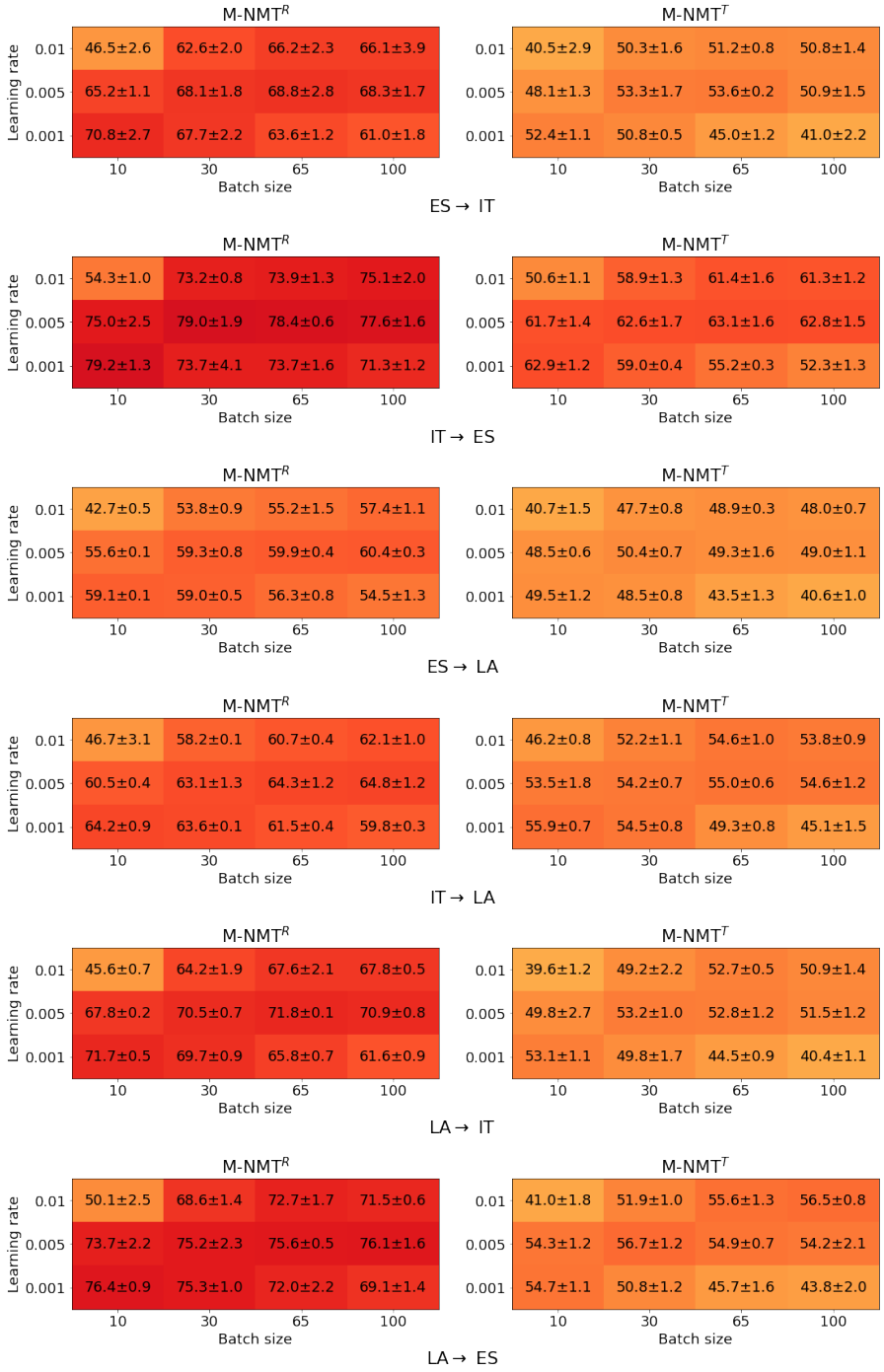


Figure C.10: BLEU results of our experiments on real data - batch size vs learning rate - 2.

C.2.3 Number of layers

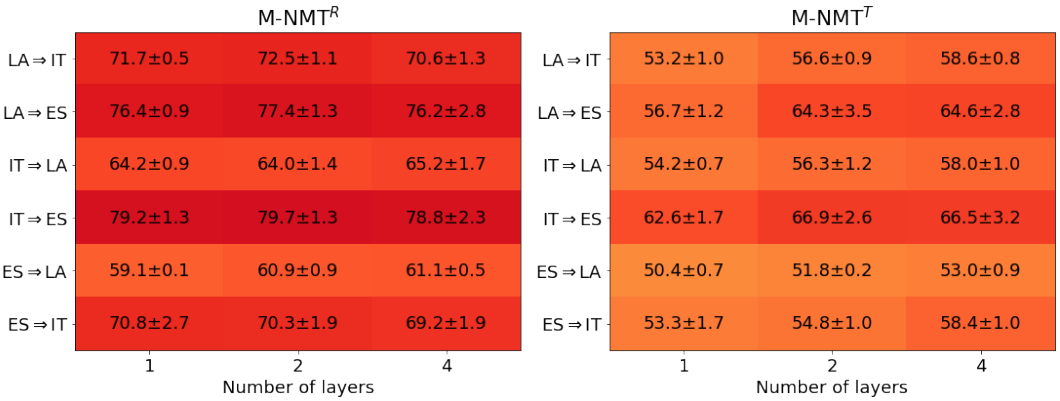


Figure C.11: BLEU results of our experiments on real data - Number of layers

C.2.4 Recurrent models: Attention type

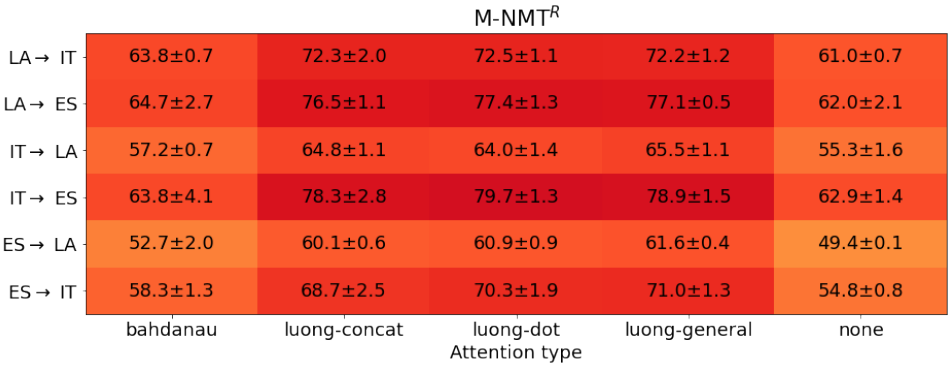


Figure C.12: BLEU results of our experiments on real data - Attention type

C.2.5 Transformers: Number of heads

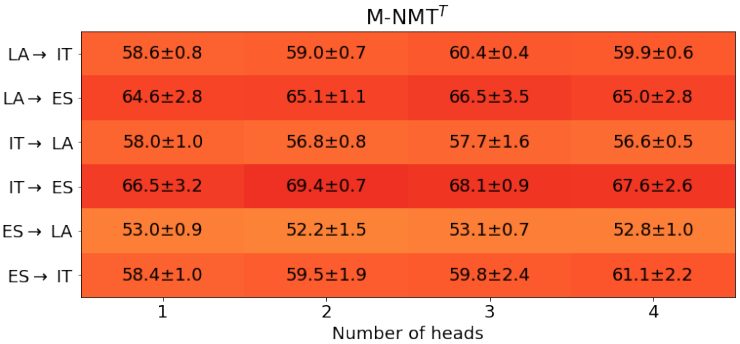


Figure C.13: BLEU results of our experiments on real data - Number of heads

D Interpretability

D.1 Complete Models BLEU Score Tables

The tables introduced here are the complete BLEU score tables for all our models language pairs, when working on real and widely multilingual data to understand the inner behavior of our different models (see Section 8.3). We provide the 1-best and 10-best predictions. The standard deviation and mean are computed across all data shufflings used to train our models. These tables therefore represent 255 models (81 language directions * 3 bilingual models * 3 shuffling seeds, + 4 multilingual models trained on all directions at once * 3 shuffling seeds).

From CA to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
1-best									
SMT	100.0 ± 0.0	72.0 ± 3.6	68.4 ± 2.3	63.4 ± 0.8	57.3 ± 0.6	85.0 ± 5.8	74.2 ± 3.0	32.6 ± 10.7	39.4 ± 3.7
B-NMT	99.6 ± 0.1	64.1 ± 3.4	45.0 ± 4.8	34.7 ± 2.3	43.9 ± 3.0	39.2 ± 7.8	52.8 ± 1.5	5.7 ± 3.0	4.8 ± 0.3
B-NMT+m	99.6 ± 0.1	74.0 ± 1.5	60.7 ± 4.6	58.4 ± 2.8	53.4 ± 2.7	77.6 ± 9.9	73.9 ± 2.9	19.9 ± 15.6	19.7 ± 8.2
M-NMT	-	64.9 ± 2.7	61.2 ± 5.9	58.7 ± 4.1	52.7 ± 1.7	63.2 ± 2.0	63.3 ± 4.5	38.4 ± 1.2	46.9 ± 5.5
M-NMT+m	89.7 ± 0.9	74.6 ± 2.4	74.5 ± 4.3	73.0 ± 2.5	58.8 ± 0.4	75.9 ± 4.3	77.2 ± 2.4	50.2 ± 11.4	49.2 ± 8.0
+shared_emb	89.2 ± 1.7	74.0 ± 0.1	73.4 ± 1.8	67.0 ± 2.7	62.1 ± 0.9	84.9 ± 5.7	77.0 ± 5.0	39.3 ± 11.6	47.3 ± 7.7
+shared_all	59.3 ± 1.3	65.0 ± 2.8	66.7 ± 4.9	62.4 ± 4.3	51.5 ± 1.7	81.2 ± 5.8	69.2 ± 3.6	45.0 ± 11.7	43.7 ± 6.5
10-best									
SMT	100.0 ± 0.0	89.8 ± 0.6	86.6 ± 2.5	81.2 ± 3.2	81.3 ± 2.3	90.2 ± 4.7	91.4 ± 2.1	63.7 ± 10.4	57.2 ± 4.5
B-NMT	99.9 ± 0.1	85.4 ± 1.2	69.9 ± 3.9	56.1 ± 3.9	64.1 ± 3.8	63.5 ± 5.3	78.3 ± 1.2	20.4 ± 9.6	12.6 ± 3.1
B-NMT+m	99.9 ± 0.1	90.2 ± 0.3	81.0 ± 2.2	76.4 ± 2.8	76.5 ± 2.7	83.8 ± 8.6	88.6 ± 2.2	35.4 ± 18.8	35.9 ± 2.9
M-NMT	-	87.1 ± 1.4	84.3 ± 3.6	79.6 ± 3.4	77.2 ± 1.6	80.6 ± 1.2	86.1 ± 3.1	63.0 ± 6.0	71.8 ± 3.6
M-NMT+m	97.8 ± 0.3	90.9 ± 1.5	89.9 ± 3.1	89.8 ± 3.1	85.7 ± 1.3	88.8 ± 4.1	92.4 ± 1.1	71.2 ± 7.9	74.5 ± 6.9
+shared_emb	97.9 ± 0.5	91.8 ± 0.9	89.6 ± 3.4	84.4 ± 3.8	88.0 ± 0.4	92.5 ± 4.5	91.7 ± 1.3	66.1 ± 6.4	77.1 ± 6.9
+shared_all	75.2 ± 1.0	87.6 ± 1.3	89.7 ± 2.0	83.9 ± 4.1	77.1 ± 2.0	93.9 ± 3.4	89.9 ± 1.6	63.8 ± 8.4	73.5 ± 10.9
From ES to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
1-best									
SMT	71.2 ± 0.4	100.0 ± 0.0	62.4 ± 0.9	67.4 ± 4.1	63.0 ± 0.5	48.6 ± 9.4	76.7 ± 2.6	34.4 ± 3.4	38.3 ± 5.5
B-NMT	73.9 ± 4.6	99.5 ± 0.1	51.6 ± 3.4	56.0 ± 3.0	57.7 ± 2.6	3.0 ± 0.2	65.9 ± 8.3	19.2 ± 5.1	5.7 ± 2.6
B-NMT+m	81.2 ± 2.9	99.5 ± 0.1	59.1 ± 4.4	69.4 ± 0.8	67.2 ± 2.2	37.8 ± 3.4	76.7 ± 2.6	26.2 ± 1.0	22.9 ± 13.1
M-NMT	72.1 ± 4.7	-	57.5 ± 2.7	70.5 ± 4.4	53.4 ± 2.5	75.7 ± 9.5	69.0 ± 3.7	37.6 ± 7.7	48.8 ± 9.0
M-NMT+m	79.0 ± 1.9	88.6 ± 1.1	67.3 ± 2.0	72.1 ± 6.2	63.1 ± 1.2	86.1 ± 3.3	73.7 ± 2.1	46.8 ± 2.7	45.9 ± 6.4
+shared_emb	80.8 ± 0.8	90.3 ± 2.5	71.4 ± 0.2	74.8 ± 2.6	64.8 ± 1.2	84.2 ± 8.0	76.4 ± 4.8	48.2 ± 5.6	42.4 ± 8.4
+shared_all	72.4 ± 3.0	61.8 ± 0.4	64.5 ± 1.9	67.3 ± 3.5	49.5 ± 3.7	78.7 ± 8.9	69.8 ± 2.5	38.2 ± 5.2	42.2 ± 8.1
10-best									
SMT	90.3 ± 1.6	100.0 ± 0.0	79.6 ± 2.5	87.2 ± 2.1	86.3 ± 0.8	78.0 ± 5.4	91.9 ± 0.9	60.4 ± 6.8	53.7 ± 7.1
B-NMT	89.3 ± 2.8	100.0 ± 0.0	69.6 ± 2.3	75.8 ± 1.1	82.7 ± 2.4	8.8 ± 1.8	85.2 ± 6.2	44.1 ± 0.8	14.0 ± 5.9
B-NMT+m	91.9 ± 1.9	100.0 ± 0.0	79.0 ± 1.0	84.7 ± 2.3	86.4 ± 2.3	60.0 ± 2.6	91.4 ± 1.1	48.3 ± 2.4	41.6 ± 8.2
M-NMT	89.9 ± 2.5	-	80.6 ± 4.2	86.5 ± 4.6	80.0 ± 2.0	92.5 ± 4.5	87.6 ± 2.0	62.4 ± 7.2	71.0 ± 6.7
M-NMT+m	93.8 ± 1.4	97.9 ± 0.4	83.8 ± 2.0	88.8 ± 3.3	86.1 ± 0.1	94.9 ± 2.5	91.5 ± 0.7	68.4 ± 6.9	69.2 ± 3.1
+shared_emb	93.9 ± 1.1	98.6 ± 0.5	85.5 ± 2.8	90.6 ± 3.1	87.2 ± 0.6	91.8 ± 6.8	93.5 ± 2.6	71.0 ± 2.9	69.3 ± 5.6
+shared_all	91.4 ± 2.0	79.9 ± 1.5	80.2 ± 4.0	88.6 ± 4.2	80.0 ± 1.8	92.6 ± 4.7	91.2 ± 0.6	64.5 ± 2.6	65.9 ± 4.5
From FR to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
1-best									
SMT	67.7 ± 2.7	63.4 ± 1.1	100.0 ± 0.0	55.9 ± 6.7	50.0 ± 3.9	32.6 ± 5.3	58.4 ± 2.9	21.5 ± 2.3	18.5 ± 6.8
B-NMT	40.1 ± 3.6	39.3 ± 5.4	98.7 ± 0.4	10.0 ± 5.8	28.9 ± 3.4	5.1 ± 0.7	31.2 ± 7.5	3.8 ± 1.5	2.3 ± 0.3
B-NMT+m	62.1 ± 3.2	58.1 ± 5.9	98.7 ± 0.4	34.3 ± 4.4	48.1 ± 5.4	7.2 ± 2.6	51.0 ± 2.1	8.4 ± 2.3	8.8 ± 2.9
M-NMT	66.0 ± 3.8	53.7 ± 2.6	-	62.8 ± 6.9	45.6 ± 3.2	62.8 ± 8.3	54.8 ± 3.5	21.8 ± 6.4	30.9 ± 19.8
M-NMT+m	74.9 ± 7.9	64.5 ± 1.5	83.8 ± 1.6	68.7 ± 4.9	53.2 ± 4.3	75.9 ± 10.8	64.8 ± 2.1	28.4 ± 3.0	21.4 ± 13.3
+shared_emb	70.9 ± 3.8	65.9 ± 4.1	81.9 ± 4.3	69.5 ± 5.6	56.3 ± 3.9	81.3 ± 10.3	65.2 ± 3.0	34.6 ± 6.4	14.5 ± 5.6
+shared_all	66.3 ± 3.8	54.0 ± 4.0	53.0 ± 5.7	57.9 ± 4.4	46.1 ± 5.4	67.3 ± 5.6	54.6 ± 2.0	28.0 ± 9.5	18.4 ± 8.8
10-best									
SMT	85.1 ± 0.9	79.9 ± 3.1	100.0 ± 0.0	72.7 ± 5.5	70.9 ± 4.4	60.1 ± 2.8	77.1 ± 2.4	32.1 ± 10.9	28.4 ± 12.7
B-NMT	59.5 ± 1.7	60.5 ± 5.8	99.2 ± 0.3	24.7 ± 5.6	49.9 ± 7.4	9.2 ± 1.2	51.4 ± 7.8	8.6 ± 1.3	9.1 ± 0.8
B-NMT+m	79.0 ± 2.6	73.2 ± 5.7	99.2 ± 0.3	55.5 ± 5.0	66.7 ± 6.1	21.1 ± 5.1	69.4 ± 1.1	15.5 ± 5.8	23.6 ± 18.4
M-NMT	83.6 ± 2.3	79.8 ± 2.0	-	82.2 ± 5.5	70.2 ± 4.4	81.4 ± 2.5	76.7 ± 2.6	46.6 ± 8.7	60.4 ± 27.2
M-NMT+m	89.8 ± 4.0	85.8 ± 1.9	94.9 ± 0.9	86.7 ± 3.0	78.8 ± 1.1	85.1 ± 12.1	82.0 ± 2.8	57.8 ± 2.5	54.4 ± 20.8
+shared_emb	89.4 ± 2.1	84.9 ± 2.3	93.3 ± 2.0	88.2 ± 5.6	76.9 ± 3.7	95.5 ± 3.2	80.7 ± 1.1	64.1 ± 7.9	42.1 ± 16.1
+shared_all	84.7 ± 3.8	76.7 ± 4.5	66.8 ± 4.2	82.2 ± 4.2	66.8 ± 3.3	92.5 ± 5.4	74.8 ± 0.7	47.0 ± 10.9	40.7 ± 14.3

Table D.1: Results of our different models for the cognate prediction task - 1/3.

From GL to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
1-best									
SMT	59.6 ± 4.1	74.9 ± 4.2	56.4 ± 9.0	100.0 ± 0.0	57.7 ± 6.6	54.6 ± 8.1	86.4 ± 1.6	29.7 ± 8.1	46.1 ± 13.6
B-NMT	38.8 ± 3.9	58.9 ± 3.0	11.4 ± 5.6	98.9 ± 1.3	30.5 ± 3.1	3.6 ± 0.5	72.7 ± 4.4	6.6 ± 1.0	4.7 ± 1.2
B-NMT+m	63.2 ± 1.7	73.2 ± 4.4	40.9 ± 7.2	98.9 ± 1.3	48.9 ± 7.7	22.6 ± 6.7	85.0 ± 0.7	15.2 ± 5.3	19.8 ± 2.2
M-NMT	69.0 ± 1.1	68.6 ± 4.7	59.6 ± 4.9	-	56.3 ± 3.8	67.9 ± 9.2	75.5 ± 1.9	45.7 ± 13.6	39.6 ± 4.6
M-NMT+m	69.4 ± 3.5	72.8 ± 2.8	64.1 ± 8.6	86.6 ± 5.0	59.8 ± 6.8	71.3 ± 14.2	82.9 ± 1.9	52.1 ± 11.0	62.3 ± 6.2
+shared_emb	72.7 ± 2.1	74.3 ± 1.5	61.5 ± 11.3	91.1 ± 1.1	62.6 ± 0.9	75.5 ± 4.0	87.1 ± 0.6	57.5 ± 11.6	57.1 ± 17.6
+shared_all	68.3 ± 3.5	68.9 ± 3.9	55.9 ± 10.7	64.2 ± 5.7	59.1 ± 5.8	69.3 ± 10.9	78.7 ± 3.9	51.0 ± 11.3	59.5 ± 4.7
10-best									
SMT	85.8 ± 0.4	89.0 ± 1.6	72.5 ± 4.4	100.0 ± 0.0	77.0 ± 5.7	78.1 ± 6.8	93.9 ± 2.2	59.3 ± 5.1	58.5 ± 14.5
B-NMT	58.9 ± 2.5	79.3 ± 2.4	22.8 ± 5.1	99.5 ± 0.6	48.3 ± 2.2	8.1 ± 2.7	87.2 ± 2.4	11.7 ± 2.3	12.6 ± 4.8
B-NMT+m	77.1 ± 1.1	87.5 ± 0.8	53.7 ± 8.3	99.5 ± 0.6	68.0 ± 6.6	48.0 ± 9.6	93.9 ± 1.0	31.1 ± 6.1	42.4 ± 7.9
M-NMT	85.3 ± 4.7	85.7 ± 2.7	76.3 ± 5.6	-	79.6 ± 5.3	89.5 ± 9.2	93.4 ± 2.0	66.3 ± 4.2	81.3 ± 3.0
M-NMT+m	89.3 ± 3.7	89.5 ± 2.4	86.4 ± 5.3	96.4 ± 2.1	82.2 ± 5.1	88.2 ± 5.3	96.4 ± 2.3	77.0 ± 4.8	85.0 ± 6.8
+shared_emb	91.1 ± 1.0	90.2 ± 2.0	84.9 ± 2.8	98.7 ± 0.6	85.3 ± 5.2	94.1 ± 1.4	95.2 ± 1.1	78.8 ± 5.5	80.9 ± 5.6
+shared_all	88.2 ± 5.3	84.7 ± 1.1	80.4 ± 6.6	85.2 ± 4.2	76.9 ± 6.3	87.3 ± 7.8	93.3 ± 2.6	68.5 ± 7.0	80.0 ± 5.0
From IT to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
1-best									
SMT	63.3 ± 3.1	74.8 ± 1.7	61.6 ± 2.8	58.2 ± 7.5	100.0 ± 0.0	44.7 ± 13.8	70.4 ± 3.1	48.6 ± 3.1	49.2 ± 0.9
B-NMT	35.5 ± 3.9	70.8 ± 0.6	31.7 ± 8.6	30.7 ± 2.9	99.6 ± 0.1	6.5 ± 5.2	61.5 ± 1.3	29.8 ± 2.6	21.9 ± 5.0
B-NMT+m	68.0 ± 0.8	73.0 ± 2.8	59.6 ± 6.4	55.2 ± 7.8	99.6 ± 0.1	35.6 ± 14.6	70.6 ± 1.5	44.7 ± 4.3	34.1 ± 4.7
M-NMT	61.0 ± 4.3	60.0 ± 4.8	55.1 ± 3.8	61.6 ± 4.0	-	55.8 ± 4.7	58.7 ± 3.5	51.9 ± 2.9	50.6 ± 3.8
M-NMT+m	73.3 ± 1.4	72.3 ± 1.7	64.3 ± 7.5	69.1 ± 5.4	81.8 ± 0.9	73.4 ± 5.8	72.9 ± 3.3	51.7 ± 2.2	52.8 ± 5.0
+shared_emb	72.8 ± 0.5	70.2 ± 3.9	66.5 ± 4.1	69.3 ± 5.4	81.4 ± 1.5	73.4 ± 8.3	73.5 ± 3.5	58.9 ± 2.8	50.9 ± 1.9
+shared_all	68.9 ± 4.1	60.8 ± 0.8	54.0 ± 6.1	59.6 ± 8.3	70.0 ± 3.3	71.2 ± 18.2	62.5 ± 2.1	44.2 ± 1.8	44.2 ± 1.1
From IT to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
10-best									
SMT	83.8 ± 2.1	89.1 ± 0.4	76.7 ± 3.0	78.3 ± 6.5	100.0 ± 0.0	68.1 ± 9.7	87.9 ± 1.7	70.2 ± 4.6	70.6 ± 1.5
B-NMT	56.0 ± 5.7	85.2 ± 1.6	53.4 ± 8.4	50.1 ± 1.5	99.9 ± 0.1	12.4 ± 4.2	83.5 ± 2.4	51.7 ± 1.7	41.2 ± 6.4
B-NMT+m	82.8 ± 0.9	87.3 ± 1.1	77.4 ± 6.4	74.8 ± 3.5	99.9 ± 0.1	51.1 ± 15.5	86.2 ± 0.6	67.2 ± 2.4	58.0 ± 3.8
M-NMT	81.8 ± 1.5	82.2 ± 3.0	76.5 ± 5.0	81.4 ± 4.4	-	79.9 ± 2.9	81.9 ± 2.1	70.5 ± 4.5	72.7 ± 4.4
M-NMT+m	90.4 ± 1.8	88.0 ± 0.6	80.0 ± 3.4	86.6 ± 2.4	96.7 ± 0.8	84.1 ± 9.8	90.0 ± 0.9	80.1 ± 1.4	73.4 ± 1.0
+shared_emb	89.6 ± 0.6	89.4 ± 1.7	80.6 ± 3.9	87.3 ± 3.1	96.5 ± 0.6	85.7 ± 9.3	89.7 ± 1.5	77.1 ± 2.0	72.2 ± 0.3
+shared_all	83.5 ± 0.7	81.3 ± 2.0	76.6 ± 7.1	80.6 ± 4.5	91.9 ± 1.6	83.3 ± 10.1	87.3 ± 1.4	71.4 ± 4.0	67.4 ± 6.9
From OC	CA	ES	FR	GL	IT	OC	PT	RO	RUP
1-best									
SMT	88.2 ± 1.8	57.8 ± 7.1	34.1 ± 5.0	57.5 ± 9.3	53.1 ± 3.0	100.0 ± 0.0	44.0 ± 6.0	21.2 ± 10.4	30.7 ± 13.8
B-NMT	60.6 ± 10.6	7.3 ± 1.1	3.4 ± 1.4	4.1 ± 2.0	8.2 ± 2.6	97.8 ± 1.1	4.0 ± 0.9	3.2 ± 1.4	4.6 ± 1.4
B-NMT+m	84.9 ± 1.2	42.4 ± 4.7	11.6 ± 6.1	19.1 ± 6.2	42.9 ± 2.5	97.8 ± 1.1	39.5 ± 7.4	10.2 ± 2.4	7.5 ± 0.3
M-NMT	75.2 ± 8.8	56.7 ± 7.8	49.1 ± 11.0	64.7 ± 8.0	55.4 ± 2.1	-	59.4 ± 2.6	47.3 ± 6.5	69.9 ± 5.5
M-NMT+m	84.8 ± 2.4	69.5 ± 4.8	54.6 ± 5.5	71.5 ± 7.4	72.0 ± 4.5	82.3 ± 6.3	59.5 ± 10.6	58.9 ± 5.6	61.1 ± 5.0
+shared_emb	86.3 ± 7.1	73.8 ± 11.2	53.5 ± 1.5	76.1 ± 13.2	69.0 ± 7.1	84.2 ± 3.8	60.0 ± 16.2	70.1 ± 13.0	74.1 ± 5.3
+shared_all	86.5 ± 2.2	60.5 ± 10.0	41.2 ± 8.7	64.7 ± 10.3	58.4 ± 6.8	59.1 ± 3.4	57.2 ± 7.8	51.3 ± 18.7	57.5 ± 11.5
10-best									
SMT	92.4 ± 2.6	80.0 ± 8.4	42.2 ± 5.3	74.0 ± 8.2	71.5 ± 2.6	100.0 ± 0.0	72.1 ± 3.4	35.9 ± 10.4	45.8 ± 6.4
B-NMT	75.2 ± 6.1	13.6 ± 3.2	8.3 ± 4.6	7.7 ± 3.5	18.6 ± 3.3	99.4 ± 0.8	8.0 ± 1.6	8.4 ± 1.9	10.4 ± 1.3
B-NMT+m	93.0 ± 2.4	63.6 ± 8.3	19.5 ± 9.8	38.0 ± 17.4	61.3 ± 1.9	99.4 ± 0.8	53.4 ± 8.5	25.1 ± 8.4	17.4 ± 4.9
M-NMT	91.0 ± 6.5	85.3 ± 6.0	61.9 ± 9.1	79.7 ± 5.5	79.5 ± 2.5	-	84.3 ± 3.9	76.4 ± 4.5	88.9 ± 11.5
M-NMT+m	94.9 ± 2.5	89.2 ± 6.0	70.5 ± 5.9	88.8 ± 6.4	88.5 ± 3.3	92.4 ± 3.1	86.7 ± 3.3	70.7 ± 4.2	88.1 ± 4.9
+shared_emb	97.1 ± 2.1	86.1 ± 7.2	67.9 ± 4.6	91.4 ± 3.2	85.6 ± 8.8	94.1 ± 1.3	86.8 ± 8.3	79.3 ± 4.0	86.0 ± 10.4
+shared_all	94.4 ± 2.4	83.1 ± 6.6	66.2 ± 5.1	85.1 ± 6.2	77.1 ± 6.2	72.0 ± 2.1	85.3 ± 3.1	71.5 ± 10.3	80.5 ± 12.3

Table D.2: Results of our different models for the cognate prediction task - 2/3.

From PT	CA	ES	FR	GL	IT	OC	PT	RO	RUP
1-best									
SMT	75.0 ± 0.1	75.4 ± 0.3	63.2 ± 5.0	89.2 ± 0.7	59.4 ± 5.9	50.8 ± 4.7	100.0 ± 0.0	42.2 ± 1.9	45.5 ± 2.3
B-NMT	66.0 ± 4.1	69.2 ± 1.0	39.0 ± 7.8	75.3 ± 3.5	50.8 ± 3.1	6.3 ± 1.6	99.3 ± 0.4	11.9 ± 5.7	10.9 ± 3.0
B-NMT+m	75.9 ± 3.0	74.9 ± 2.1	56.2 ± 2.7	86.0 ± 2.1	59.5 ± 4.2	29.2 ± 5.9	99.3 ± 0.4	28.8 ± 6.8	27.3 ± 3.8
M-NMT	74.0 ± 3.3	69.2 ± 2.3	63.9 ± 3.6	77.2 ± 0.3	55.4 ± 3.7	72.4 ± 6.6	-	48.8 ± 6.4	62.1 ± 5.6
M-NMT+m	78.7 ± 3.9	75.8 ± 4.0	67.8 ± 0.5	83.9 ± 1.7	63.8 ± 1.6	89.1 ± 3.3	89.0 ± 1.7	55.7 ± 5.9	61.0 ± 12.6
+shared_emb	78.0 ± 3.4	73.1 ± 2.9	70.3 ± 4.1	82.2 ± 3.0	61.4 ± 2.3	81.9 ± 5.7	88.4 ± 1.9	52.9 ± 7.2	61.7 ± 3.2
+shared_all	76.4 ± 3.0	67.3 ± 0.7	63.4 ± 3.6	78.0 ± 3.7	55.1 ± 2.9	71.2 ± 5.4	64.2 ± 2.2	47.7 ± 5.6	56.1 ± 8.8
10-best									
SMT	86.9 ± 1.1	91.6 ± 0.7	83.1 ± 4.9	96.2 ± 1.0	80.9 ± 3.6	76.4 ± 9.6	100.0 ± 0.0	67.8 ± 4.8	74.2 ± 2.1
B-NMT	80.1 ± 3.3	88.5 ± 0.5	61.0 ± 5.2	89.1 ± 2.3	73.6 ± 2.5	11.7 ± 1.5	99.8 ± 0.1	24.2 ± 1.3	36.5 ± 3.3
B-NMT+m	86.5 ± 2.7	89.5 ± 0.8	76.0 ± 4.0	93.9 ± 1.6	82.0 ± 3.6	43.6 ± 3.7	99.8 ± 0.1	43.2 ± 4.6	51.3 ± 2.4
M-NMT	88.5 ± 2.2	89.0 ± 1.4	85.8 ± 2.8	93.0 ± 1.1	80.0 ± 3.6	90.4 ± 2.4	-	70.3 ± 5.9	83.8 ± 2.2
M-NMT+m	90.0 ± 3.1	92.1 ± 1.0	86.6 ± 3.0	94.5 ± 1.9	85.1 ± 2.1	96.4 ± 4.3	98.7 ± 0.7	77.8 ± 4.2	80.3 ± 11.6
+shared_emb	89.7 ± 2.8	91.4 ± 1.0	89.0 ± 2.6	95.8 ± 1.3	85.2 ± 2.8	95.4 ± 3.9	97.7 ± 1.1	73.6 ± 9.8	84.4 ± 3.2
+shared_all	87.0 ± 1.1	88.6 ± 2.3	85.5 ± 1.9	92.9 ± 1.3	75.9 ± 2.1	93.1 ± 4.2	84.6 ± 3.0	69.6 ± 2.8	85.0 ± 1.5
From RO to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
1-best									
SMT	32.9 ± 5.3	37.6 ± 5.8	20.2 ± 2.6	29.7 ± 10.4	43.5 ± 5.6	25.5 ± 2.8	32.7 ± 0.6	100.0 ± 0.0	66.3 ± 1.7
B-NMT	10.4 ± 3.4	22.6 ± 4.5	6.3 ± 1.4	2.1 ± 0.5	33.1 ± 8.7	7.1 ± 2.5	14.9 ± 6.2	98.5 ± 1.4	59.0 ± 8.3
B-NMT+m	18.1 ± 4.8	34.2 ± 3.0	7.2 ± 3.3	15.9 ± 2.5	44.7 ± 7.0	12.9 ± 1.8	21.7 ± 7.3	98.5 ± 1.4	67.4 ± 9.8
M-NMT	47.9 ± 2.4	48.5 ± 2.1	37.9 ± 9.2	47.0 ± 3.9	42.0 ± 6.6	51.0 ± 17.3	42.0 ± 5.6	-	58.1 ± 8.3
M-NMT+m	47.2 ± 4.0	56.4 ± 7.4	36.9 ± 10.7	55.6 ± 4.1	53.2 ± 2.2	59.1 ± 13.5	45.7 ± 3.4	70.4 ± 2.3	70.7 ± 9.4
+shared_emb	57.7 ± 7.1	54.2 ± 3.7	36.0 ± 4.7	54.6 ± 6.6	55.1 ± 4.9	63.0 ± 13.4	50.7 ± 6.3	70.4 ± 1.8	75.6 ± 8.0
+shared_all	53.7 ± 5.3	33.1 ± 5.6	37.8 ± 6.3	50.9 ± 6.8	37.3 ± 2.2	56.8 ± 11.5	38.4 ± 7.3	48.1 ± 0.9	63.4 ± 7.4
10-best									
SMT	57.9 ± 3.9	63.7 ± 7.6	38.1 ± 6.1	47.0 ± 6.4	72.1 ± 4.2	44.5 ± 9.6	58.3 ± 2.3	100.0 ± 0.0	87.4 ± 2.0
B-NMT	22.5 ± 10.8	45.4 ± 0.7	10.0 ± 0.4	6.0 ± 0.2	58.1 ± 5.6	14.2 ± 3.8	30.8 ± 4.8	99.6 ± 0.5	80.8 ± 9.8
B-NMT+m	38.2 ± 8.4	58.3 ± 4.5	16.2 ± 5.2	32.9 ± 8.8	64.9 ± 4.3	27.2 ± 3.9	51.5 ± 4.0	99.6 ± 0.5	85.7 ± 8.8
M-NMT	79.6 ± 4.8	75.7 ± 5.9	56.6 ± 16.0	66.9 ± 2.0	71.3 ± 4.5	74.7 ± 15.3	70.2 ± 3.7	-	80.1 ± 9.2
M-NMT+m	75.9 ± 5.6	80.5 ± 8.0	52.8 ± 8.8	76.2 ± 5.9	80.8 ± 3.9	77.9 ± 7.5	75.8 ± 3.7	89.3 ± 3.3	87.2 ± 4.9
+shared_emb	80.8 ± 5.5	82.7 ± 4.6	65.2 ± 6.1	81.0 ± 5.3	82.4 ± 2.1	83.0 ± 14.0	76.0 ± 2.4	89.5 ± 1.0	90.2 ± 7.0
+shared_all	74.6 ± 9.8	64.5 ± 6.2	60.8 ± 7.1	69.7 ± 8.4	67.0 ± 3.3	66.9 ± 14.3	68.3 ± 4.6	64.5 ± 1.6	84.8 ± 6.3
From RUP to	CA	ES	FR	GL	IT	OC	PT	RO	RUP
1-best									
SMT	29.2 ± 2.4	32.4 ± 1.9	21.7 ± 2.9	29.5 ± 13.2	36.6 ± 4.1	26.1 ± 12.4	42.0 ± 5.5	63.3 ± 7.3	100.0 ± 0.0
B-NMT	2.7 ± 0.7	3.3 ± 0.7	5.7 ± 1.0	3.1 ± 1.9	26.7 ± 2.6	5.2 ± 2.0	27.1 ± 3.0	48.8 ± 5.1	95.2 ± 1.8
B-NMT+m	16.4 ± 4.5	23.4 ± 1.9	9.1 ± 1.7	15.4 ± 9.1	30.6 ± 0.4	14.4 ± 5.3	28.9 ± 12.6	64.8 ± 5.4	95.2 ± 1.8
M-NMT	50.1 ± 12.7	36.7 ± 6.3	32.0 ± 12.7	33.4 ± 1.9	44.4 ± 4.9	29.9 ± 3.1	56.8 ± 5.6	57.7 ± 3.0	-
M-NMT+m	60.0 ± 4.8	51.8 ± 7.4	24.6 ± 14.4	49.6 ± 8.0	44.7 ± 3.5	63.5 ± 7.9	60.4 ± 7.1	67.9 ± 4.7	70.4 ± 6.0
+shared_emb	59.2 ± 8.4	47.2 ± 3.5	46.7 ± 5.0	54.6 ± 6.7	48.9 ± 4.3	41.7 ± 11.4	61.6 ± 5.6	66.7 ± 3.4	75.6 ± 3.2
+shared_all	46.9 ± 20.6	25.1 ± 6.7	35.2 ± 18.3	37.3 ± 12.1	34.0 ± 5.0	53.6 ± 9.9	39.0 ± 12.7	52.6 ± 6.4	59.8 ± 2.1
10-best									
SMT	53.8 ± 14.2	60.4 ± 7.7	32.4 ± 11.5	45.7 ± 6.8	62.6 ± 0.7	35.2 ± 11.3	62.7 ± 9.1	83.1 ± 7.4	100.0 ± 0.0
B-NMT	8.3 ± 4.1	15.6 ± 6.8	13.4 ± 3.8	7.0 ± 2.3	44.6 ± 2.0	7.3 ± 1.0	46.6 ± 5.3	72.0 ± 7.0	98.4 ± 1.3
B-NMT+m	25.1 ± 6.0	51.8 ± 4.7	17.8 ± 5.4	22.1 ± 10.9	51.9 ± 1.8	31.1 ± 15.8	51.8 ± 9.9	80.9 ± 8.1	98.4 ± 1.3
M-NMT	77.4 ± 9.0	72.3 ± 1.5	62.2 ± 11.2	66.9 ± 8.0	69.4 ± 6.0	46.7 ± 12.2	79.0 ± 1.5	79.5 ± 0.7	-
M-NMT+m	73.6 ± 10.6	80.1 ± 6.5	53.4 ± 18.4	78.7 ± 12.1	72.5 ± 3.3	77.2 ± 7.1	81.6 ± 5.6	83.2 ± 4.0	89.2 ± 4.4
+shared_emb	79.2 ± 12.5	78.6 ± 11.0	63.4 ± 9.6	77.6 ± 7.2	74.7 ± 1.9	82.3 ± 3.3	80.8 ± 1.5	83.4 ± 5.6	89.9 ± 3.2
+shared_all	69.1 ± 13.9	60.9 ± 7.2	62.4 ± 14.9	62.3 ± 1.5	64.0 ± 3.4	73.6 ± 18.8	72.9 ± 4.8	77.9 ± 8.4	76.4 ± 0.9

Table D.3: Results of our different models for the cognate prediction task - 3/3.

D.2 Consonant space t-SNE

This section introduces the t-SNE of the consonant space, to provide a new representation of the dimension reduction described in Section 8.5.2.2. We still observe a phonetically significant division of the space in the different dimensions.

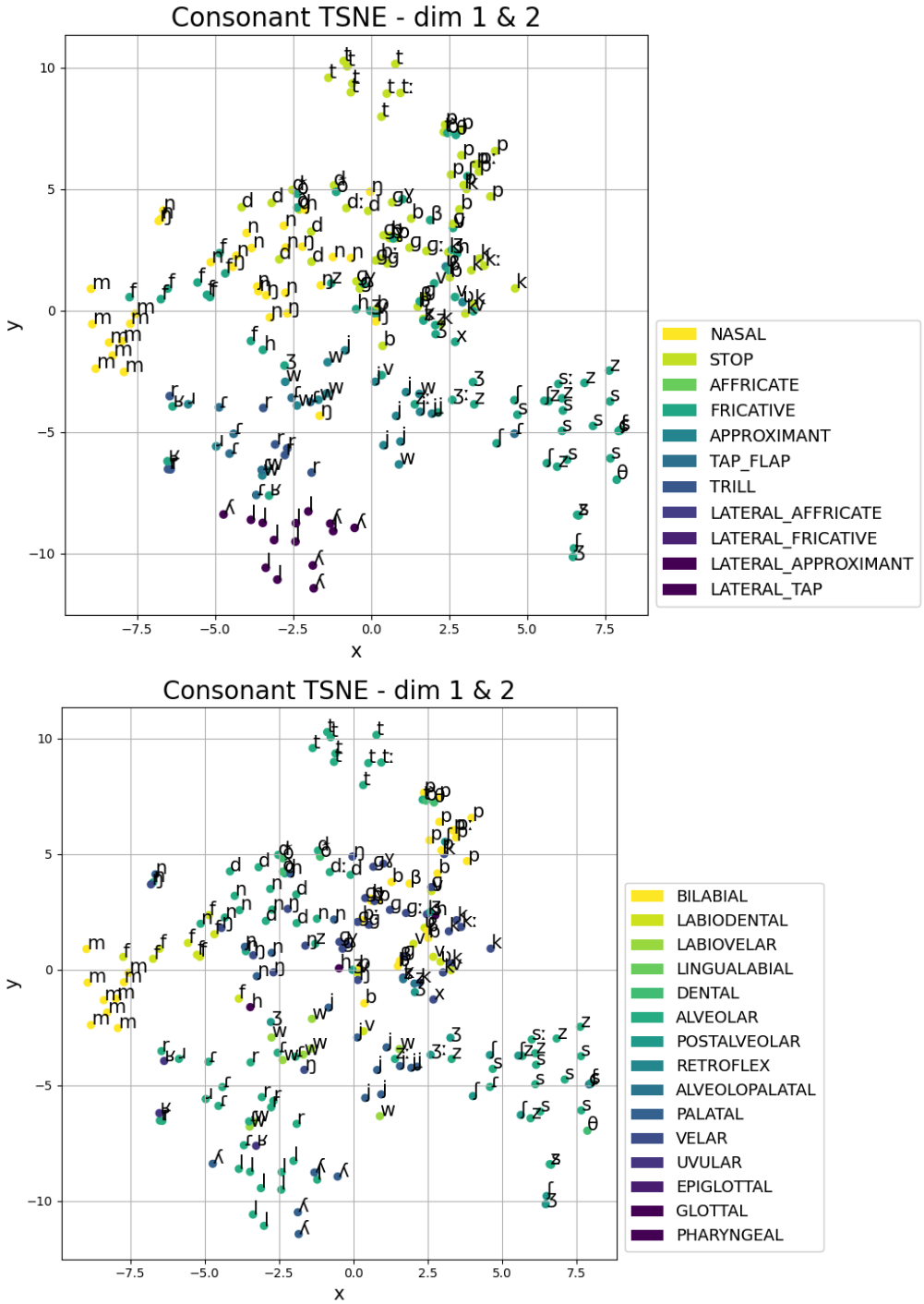


Figure D.1: Consonant t-SNE, seed 0, coloured on manner above and on place below

D.3 BLEU frequencies

We display the BLEU frequency histograms for the remaining languages pairs (following Section 8.3.6), to better understand the repartition of correctly predicted words (high BLEU) vs badly predicted words (low BLEU) depending on the language and model type.

Figure D.2: BLEU frequencies, from all our languages to Galician, Italian, Occitan.

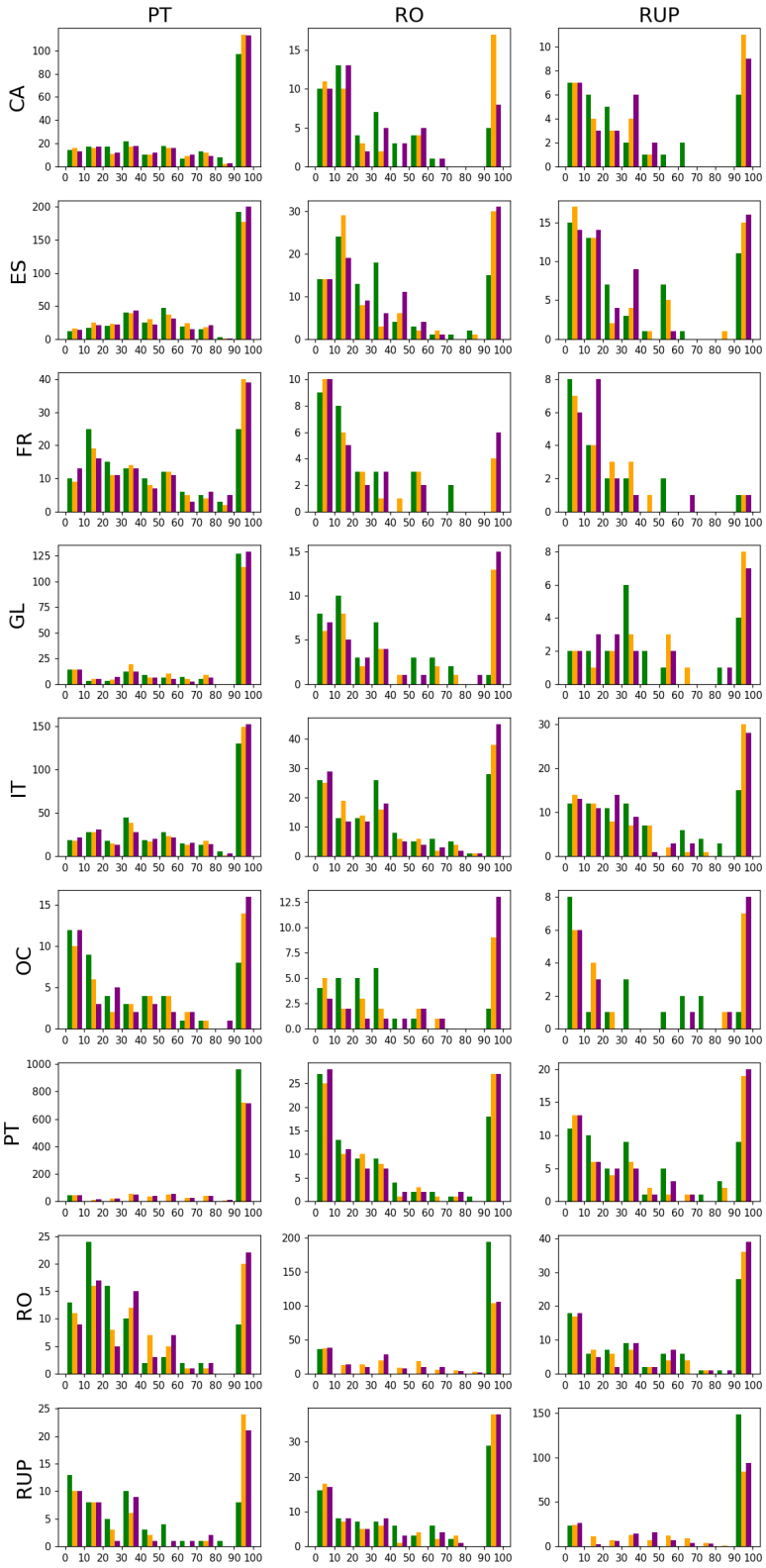


Figure D.3: BLEU frequencies, from all our languages to Portuguese, Roumanian, Aromanian.

D.4 Phone accuracy and frequency

We display the phone accuracy of our three language pairs of interest (ES-PT, PT-GL, RO-FR) as a function of the phone frequency, as mentioned in Section 8.4.2.

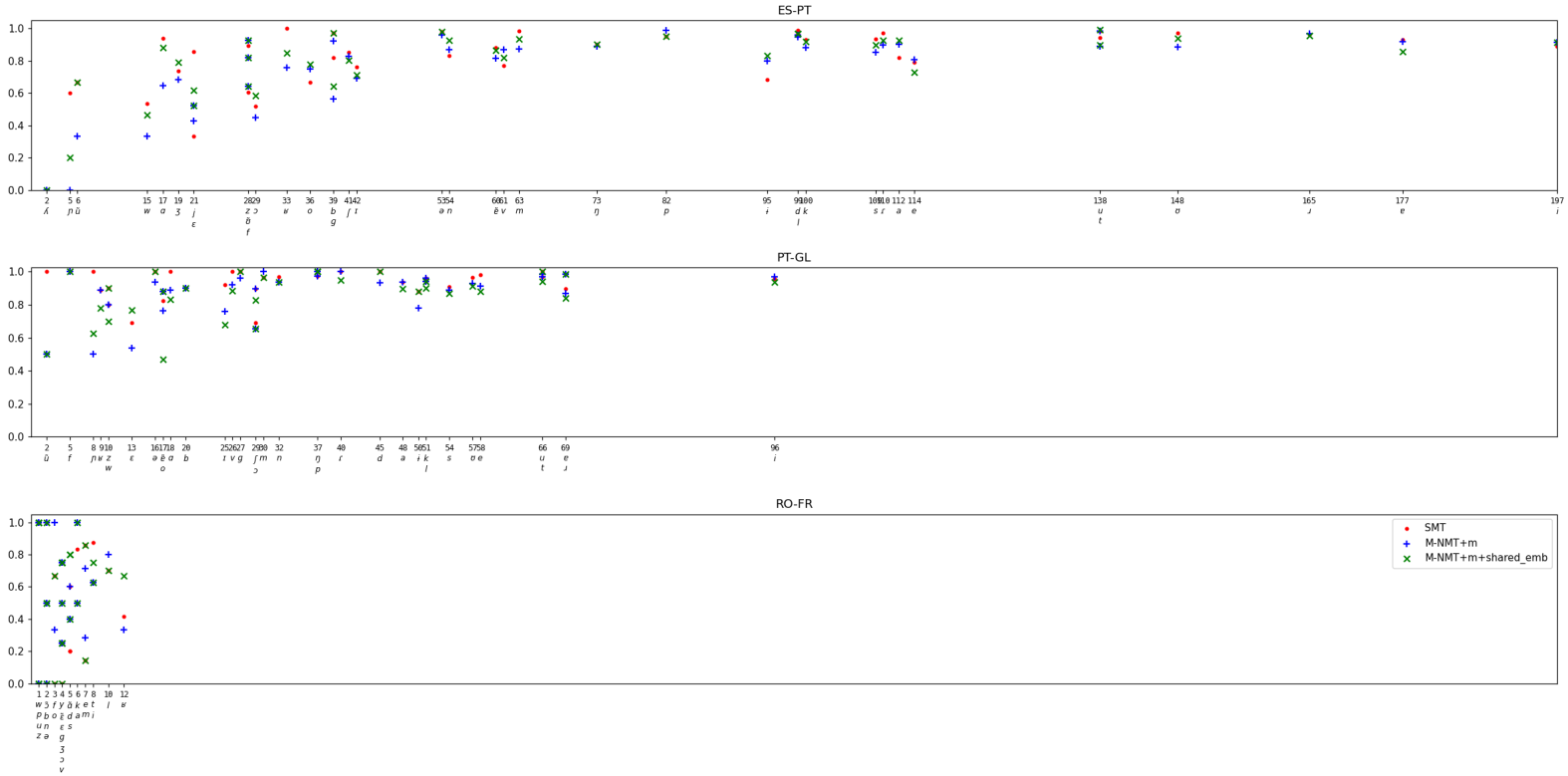


Figure D.4: Matching accuracy as a function of character frequency (ES-PT, PT-GL, RO-FR).

D.5 Raw phones correspondence predicted

In this section, we look at the raw phones that our best models (SMT, M-NMT+m, M-NMT+m+shared_emb) predicted for each language pair, when studying Romance sound correspondences by looking at French, Italian, Spanish, Portuguese, and Romanian (see Section 8.6.1.1). We compare them to the "gold" truth, the phones most frequently aligned in the target language to the input phone, obtained through the Needleman-Wunsch alignement algorithm with our custom phonetic cost function (see Appendix D.5.6).

D.5.1 From Spanish

Spanish to	Italian			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, aː, da	a, aː	a, aː	a(72%), aː(26%)
b	b, v, ve	b, f, v	b, bː, f, p, v	v(58%), b(38%)
d	d, de, di, dine, do, te	d	d, f	d(93%), t(6%)
e	dʒe, e, re, ɛ	a, e, eː, f, i, l, s, v, ɛ, ɛː	a, e, eː, i, j, ɛ, ɛː	e(62%), ɛ(13%), i(8%), ɛː(6%)
f	f, ff	f	f	f(99%)
i	dʒi, e, i, iko	i, iː	i, iː	i(70%), iː(20%), e(6%)
j	i, j, ri	e, eː, i, iː, s, ɛ	a, d, e, f, i, j, l, v, ɛ, ɛː	i(57%), j(11%), s(9%), ʃ(9%)
k	k, kr, kw	k	k	k(89%), kː(6%)
l	l, le, lle, lo	l	l	l(94%)
m	m, mm, n	m	m	m(92%), n(8%)
n	n, ne, no	n	a, m, n	n(97%)
o	do, o, ro, vo	f, o, uː, v, w, ɔ, ɔː, ʊ	o, oː, u, ɔ, ɔː, ʊ	o(88%)
p	p	p	p	p(93%)
r	or, r, rr	r	r	r(100%)
s	s, se, so, ss, sse	s	e, o, s	s(77%), z(21%)
t	t	t	t	t(91%), tː(8%)
u	bo, dʒu, o, otːo, u, uo, uː, ʊ	f, i, u, uː, ʊ	f, u, uː, ʊ	ʊ(40%), uː(19%), u(18%), o(17%)
w	fɔ, o, w, ɔ, ʊ	f, o, u, uː, w, ɔ, ʊ	aː, f, k, kː, o, u, w, ɔ, ɔː, ʊ	w(40%), ʊ(32%), uː(8%)
x	dʒ	b, d, f, l, p, s, v, ʎ	d, e, j, k, l, r, s, v, ɛ, g	ʒ(61%), d(12%), ʎ(11%), s(9%)
ð	d, di, t, tr	d, v	d, e, f, i, t	d(57%), t(40%)
ŋ	n, ni, ŋ	f, n, v, ŋ, g, ɲ	a, b, e, m, n, o, p	ŋ(71%), n(29%)
ɔ	do, o, ɔ, ɔː	f, o, oː, uː, v, w, ɔ, ɔː, ʊ	o, oː, u, ɔ, ɔː, ʊ	o(45%), ɔː(19%), oː(16%), ɔ(16%)
ɛ	de, dʒe, e, ɛ	a, e, eː, i, iː, l, o, v, ɛ, ɛː	a, e, eː, i, j, ɛ, ɛː	e(46%), ɛ(30%), ɛː(8%), i(5%)
g	g, gw, gwi	b, f, k, v, g	f, k, g	g(81%), ɲ(11%)
ɣ	k, kw, g	b, f, k, p, g	f, k, g	g(68%), k(31%)
ɾ	dʒːe, i, j	d, e, f, i, l, o, s, v, w	a, aː, d, e, i, iː, j, ɛ, ɛː, g	i(33%), iː(33%), e(33%)
ɲ	ne, nn, ɲ	m, n, ŋ, ɲ	a, b, e, m, n, p	ɲ(59%), n(41%)
ɾ	dere, re	r	r	r(97%)
ʃ	l, m, uː, ʃ, ʒ	d, f, i, l, p, s, t, tː, ʊ	e, eː, o, r, s, ɔ, ɛ, ʃ	t(40%), ʃ(40%), ʒ(20%)
ʊ	dʊ, l, li, p, pi, s, ʊ	b, f, l, p, v	b, f, o, oː, u, uː, ɔ, ɔː, ʊ	ʊ(77%), i(15%)
ʎ	kj, l, ll, pj	l	d, l, g	l(79%), j(16%), ʎ(5%)
ʝ	dʒ	d, e, eː, f, i, iː, l, s, v, ɛ	a, aː, d, e, f, i, iː, j, v, ɛː	ʒ(83%), j(17%)
β	b, p, s, v	b, f, v	b, bː, f, k, p, v	v(51%), b(34%), p(8%)
θ	altʃe, tʃe	s, t	s, t	ʃ(42%), s(28%), t(16%)

Table D.4: Phones predicted by our different models with a confidence above 0.05, from Spanish to Italian. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

Spanish to	Portuguese			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, ɐ	a, ɐ, ɐ̃	a, ɐ	ɐ(51%), a(35%), ɐ̃(8%)
b	b, v	v	b, v	v(60%), b(38%)
d	dɨ	d	d, f	d(99%)
e	dɨ, e, eɪ, ɛ, ɨ	e, i, ɛ, ɨ	e, i, ɛ, ɨ	ɨ(41%), e(39%), ɛ(9%), i(6%)
f	f	f	f	f(100%)
i	i	i	i	i(95%)
j	i, j, ɐ̃	e, f, i, j, v, ɛ, ɨ, ʒ	d, e, i, j, ɛ, ɨ	j(45%), i(37%), e(9%)
k	k	k	k	k(97%)
l	l, lɨ	l	l	l(90%), w(6%)
m	m	m	m	m(92%), ŋ(7%)
n	n, ŋ, ɱ, ɐ̃	n, ɐ̃, ɪ	n, ŋ	ŋ(45%), n(36%), ɐ̃(14%)
o	u, ʊ	f, o, u, ũ, v, ɔ	o, u, ɔ	ʊ(67%), u(21%), o(7%)
p	p	p	p	p(98%)
r	ʁ	ʁ	e, ɹ, ɾ, ʁ	ʁ(99%)
s	s, sɨ, ʃ	s, ʃ	s, z, ʃ	s(45%), ʃ(32%), z(20%)
t	t	t	t	t(99%)
u	fu, u	f, u, ũ	u	u(82%), ũ(12%)
w	o, u, w, ɔ	f, o, u, ũ, ɔ	f, k, o, u, ɔ	u(59%), w(35%)
x	ʒ	s, ʒ	d, k, s, v, g, ʃ, ʒ	ʒ(85%), ʎ(8%)
ð	d	d	d, f	d(97%)
ŋ	ŋ, ɱ	k, n, g, ʒ	a, b, e, m, n, p, ɐ, ɐ̃, g, ɲ	ŋ(100%)
ɔ	foɪ, o, ow, u, ɐ̃, ɐ̃ʊ̃, ɔ, ʊ	f, o, u, ũ, ɔ	o, u, ɔ	u(31%), ɔ(21%), ɐ̃(20%), o(20%), ʊ(7%)
ɛ	e, eɪ, ɛ, ɨ	e, i, v, ɛ, ɨ	e, i, ɛ, ɨ	e(53%), ɨ(24%), ɛ(12%)
g	g, gw	k, g, ʒ	k, g	g(98%)
ɣ	g	k, g, ʒ	k, g	g(98%)
ɪ	i, ɪ	d, f, i, k, s, v, g, ʒ	e, i, j, ɐ, ɛ, g, ɨ, ʒ	ɪ(86%), i(14%)
ɲ	n, ɪ, ɲ	n, g, ɪ, ɲ	b, e, m, n, p, ɐ, ɐ̃, g, ɲ	ɲ(48%), n(30%), ŋ(9%)
ɾ	ɹ, ɾ	d, ɹ, ɾ, ʁ	a, ɹ, ɾ, ʁ	ɹ(62%), ɾ(37%)
ʃ	iŋ, m, ɪ	k, s, ʃ, ʒ	k, o, p, s, z, ɔ, g, ɨ, ʃ, ʒ	ʃ(75%), t(25%)
ʊ	ʊ	b, f, k, o, p, s, u, v, ɔ, ʒ	b, f, k, o, t, u, ũ, ɔ, ʊ	ʊ(85%), u(10%)
ʎ	l, ʃ	l, ʃ	e, l, ɐ, ɔ, g, ɨ, ʃ	l(76%), ʃ(16%)
ʝ	i, ʒ	d, e, f, i, s, v, ɛ, g, ɨ, ʒ	d, e, f, i, j, ɛ, g, ɨ, ʒ	ʒ(67%), i(33%)
β	b, v, vɨ	b, v	b, k, v	v(57%), b(42%)
θ	s, sɨ, ʃ	s	k, s, t	s(81%), z(7%), ʃ(6%)

Table D.5: Phones predicted by our different models with a confidence above 0.05, from Spanish to Portuguese. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

Spanish to	French			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, aak, aaks, ak, aʁ	a	a	a(69%), e(14%), ɑ̃(7%)
b	b, v	b, f, p, v	b, v	v(50%), b(48%)
d	d	d	d	d(100%)
e	asjɔ̃, e, jasjɔ̃, je, ə, ɛ	e, i, s, ɛ, ʃ	e, i, j, ɛ	e(44%), ɛ(31%), ə(8%)
f	f	f	f	f(98%)
i	i	i	i	i(82%), ɛ̃(13%)
j	i, j, ʒ	e, i	a, e, i, j	j(79%), ʁ(10%)
k	aʁtik, k, tik	k	k	k(96%)
l	l, li	l	l	l(97%)
m	m	m	m	m(98%)
n	n	n	n	n(91%)
o	e, k, o, s, u, y, ɔ, ɔn, ə-	u, y, ɔ	ɔ	ɔ(52%), o(8%), ø(7%), y(6%)
p	p	p	p	p(99%)
r	ʁ	ʁ	ʁ	ʁ(100%)
s	s	s	s	s(79%), z(15%)
t	t	t	t	t(93%)
u	fy, y	y	y	y(88%), ɔ̃(8%)
w	y, ɔ	y	a, k, o, y, œ̃, ɔ	y(40%), w(20%), v(20%), s(20%)
x	ʒ	f, s, ʒ	s, ʒ	ʒ(69%), s(9%), k(9%), l(6%)
ð	d, t	d	d	d(57%), t(33%), ʁ(7%)
ŋ	ŋ	f, k, g, ʒ	m, n	
ɔ	o, œ, ɔ, ɔ̃	y, ɔ, ɔ̃	o, u, y, ɔ	ɔ(45%), ɔ̃(42%), œ(7%)
ɛ	e, ɛ	e, i, s, v, ə, ɛ, ʒ	a, e, i, j, ɛ	ɛ(35%), ɑ̃(33%), e(26%)
g	k, g, ʒ	f, k, g	k, g	g(62%), n(19%), k(12%), ʒ(6%)
ɣ	g, ʒ	f, k, g, ʒ	g, ʒ	g(86%), ʒ(9%)
ɪ	gzism	d, f, i, k, v, ɛ̃, g, ʃ, ʒ	a, d, e, i, j, ɑ̃, ə, ɛ, ʒ	i(100%)
ɲ	ɲ	n, ɲ	m, n, ɑ̃, ɲ	ɲ(67%), n(33%)
ɾ	aʁ, ʁ, ʁal, ʁatif	ʁ	ʁ	ʁ(97%)
ʃ	k	f, k, s, g, ʃ, ʒ	k, s, ʃ, ʒ	t(33%), l(33%), ʃ(33%)
ʊ	b, p	b, f, k, m, p, u, v, y, ɔ	a, b, f, k, o, u, y, œ̃, ɔ	o(73%), p(9%), k(9%), b(9%)
ʎ	l	b, l, p	l, ʒ	l(71%), w(14%), n(14%)
ʝ	ʒ	d, e, f, i, j, v, ɛ, ʃ, ʒ	a, d, e, i, j	ʒ(100%)
β	b, f, v, vj, vja	b, f, p, v	b, v	b(56%), v(32%), f(10%)
θ	s	s	s	s(94%)

Table D.6: Phones predicted by our different models with a confidence above 0.05, from Spanish to French. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

Spanish to	Romanian			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, ra, ə	a	a	a(47%), ə(36%), ɨ(7%), e(6%)
b	b, v	v	b, v	v(61%), b(33%)
d	d, t, te	d, z	d, z	d(62%), t(22%), z(9%)
e	a, dʒea, e, ea, rə, ə, ʒe	a, e, i, j, o, s, ʒ	a, e, i, j, ə	e(58%), a(11%), o(8%), ə(8%), i(6%)
f	f	f	f	f(100%)
i	e, i, j, je, ə, ʒi	j	a, d, e, f, i, j, z	i(58%), e(21%), j(6%), ə(6%)
j	e, f, i, j, s	e, j, v, ʃ, ʒ	a, d, f, j	j(39%), i(17%), s(13%), g(9%), ʃ(9%)
k	k, kr	k	k	k(94%)
l	l, r	l	l	l(65%), ɾ(18%), r(12%)
m	m	m	m	m(95%), n(5%)
n	n, ne, nə	n	n	n(94%)
o	a, aʊ, fu, na, o, u, ur, ɔa, ʊ, ʒe	a, o, u	e, o, u, ɔ, ə	u(44%), ʊ(14%), o(13%), e(6%), ɔ(5%), ə(5%)
p	p	p	p	p(100%)
r	r	r	r	r(92%), ɾ(8%)
s	s, se	s, ʃ	s	s(83%), ʃ(11%)
t	st, t	p, s, t	t	t(88%), p(6%)
u	fu, u	a, f, u	u, w, ɔ	u(77%), o(10%), ʊ(6%)
w	fɔ, u, w, ɔ, ə	a, o, u, ɔ	a, j, o, u, ɔ, ə	ɔ(44%), p(22%), w(6%), v(6%), o(6%), f(6%), b(6%), ʃ(6%)
x	dʒ, ls, s, ʒ	d, f, h, l, s, ʒ	j, k, s, v, ʃ, ʒ	s(45%), ʒ(36%), l(9%), ʃ(9%)
ð	d, t	d, j, v, z	d, z	t(61%), d(29%)
ŋ	m, n, ne, ŋ	d, f, n, ŋ, g, ɨ	d, m, n, p, ŋ, ɨ	ŋ(60%), n(20%), m(20%)
ɔ	e, fu, o, u, ɔa	a, o, u, ɔ	o, u, ɔ	u(58%), o(17%), ɔ(11%), e(8%)
ɛ	a, e, i, j, je	a, e, i, j, o, v, ʒ	a, d, e, i, j, ə, ʒ	e(47%), a(16%), i(15%), ɨ(8%)
ɡ	k, g, gr	d, f, k, g, ʒ	k, g	g(77%), k(15%), ʒ(8%)
ɣ	k, p, g	d, k, g	d, f, k, g	k(68%), g(32%)
ɪ	dʒe, jept	d, j, v, z, ʃ, ʒ	a, d, e, j, p, s, t, z, ə	e(67%), ə(33%)
ɲ	m, n, ɪ	m, n, ɨ	d, m, n, p, v, ɨ	n(36%), m(27%), ŋ(27%), ɪ(9%)
ɾ	a, r	r	r	r(67%), ɾ(19%)
ʃ	k, p, ʃ	s	r, s, ʃ	t(62%), ʃ(25%), k(12%)
ʊ	u	a, d, f, h, l, o, s, u, v, ʒ	a, j, o, u, ɔ, ə, ʊ	u(100%)
ʎ	k, l, p, pl	f, l, p, s	l	l(75%), w(8%), p(8%), k(8%)
ʝ	d, j	d, e, f, j, v, ʃ, ʒ	a, d, f, j, z, ʒ	j(67%), ʒ(33%)
β	b, p, v, ʊ	v	b, v	b(45%), p(35%), v(10%)
θ	de, tʃ, tʃe	s, ʃ	s, t, ʃ	ʃ(59%), t(19%), s(6%), ʒ(6%)

Table D.7: Phones predicted by our different models with a confidence above 0.05, from Spanish to Romanian. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

D.5.2 From Italian

Italian to	Spanish			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
extipaa	a, ad, aɾ	a	a	a(97%)
aː	a	a	a	a(98%)
b	b, β	b, p	b	β(59%), b(41%)
bː	p, β, βl	b, p	b, f, β	β(80%), p(20%)
d	d	d, t, ʝ	d	d(47%), ð(44%)
dː	bð	b, d	d, p, t	b(100%)
e	bre, e, es, ɛ, ɛɾ	e, ɛ	e, ɛ	e(61%), ɛ(29%), i(7%)
eː	e, ɛ	e, i, ɛ	a, e, ɛ	e(69%), ɛ(25%), i(6%)
f	f	f	f	f(92%)
i	e, i, in	e, i	e, i	i(67%), j(20%), e(7%)
iː	i	e, i	e, i	i(91%)
j	l, le	e, f, i, j, u, ɛ	e, i, j, l, ɛ	j(66%), l(20%), ʎ(7%)
k	k	k	k	k(92%), ɣ(7%)
kː	ek, k	k	k	k(96%)
l	el, l	l, ʎ	l, ʎ	l(92%), ʎ(6%)
m	m	m	m	m(98%)
n	n	n	n	n(91%)
o	o	o, u, ɔ	o, u, w, ɔ	o(76%), ɔ(17%)
oː	o, ɔ	o, u, ɔ	o, w, ɔ	ɔ(57%), o(38%)
p	p	p	p	p(94%), β(6%)
pː	p	p	p	p(100%)
r	r, ɛɾ, ɾ	r, ɛ, ɾ	r, ɾ	ɾ(89%), r(9%)
s	eks, es, s	s	s	s(77%), θ(14%)
sː	sː	s	p, s, x	θ(100%)
t	t	t	t	t(79%), ð(12%), θ(6%)
tː	kt, t, tj, tʃ	k, t	t	t(91%), ð(7%)
u	o, u, ua	o, u	o, u, ɔ	u(91%), ɔ(6%)
uː	o, u	o, u	o, u, ɔ	u(87%)
v	b, β	b, f, β	b	β(62%), b(36%)
w	bɾ, u, w, ɲa	a, e, f, i, j, o, u, w, ɔ	a, k, l, o, u, w, ɔ, ʊ	w(54%), j(25%), ɾ(8%)
z	ks, s	s	s	s(95%)
ŋ	d, ŋ	n	b, m, n, p, s, ŋ, g, ɲ	ŋ(94%), n(6%)
ɔ	o, ɔ, ɔn	o, u, w, ɔ	o, u, w, ɔ	ɔ(61%), o(27%), e(7%)
ɔː	o, we, ɔ, ɔn, ɛo	o, u, w, ɔ	o, w, ɔ	ɔ(67%), o(19%), ɛ(7%), e(6%)
ɛ	e, es, ɛ	e, ɛ	e, ɛ	ɛ(55%), e(40%)
ɛː	e, ɛ	e, ɛ	a, e, ɛ	e(54%), ɛ(46%)
g	g, ɣ	k, g, ɣ	k, g, ɣ	ɣ(60%), g(36%)
gː	ɣ	k, g, ɣ	k, g, ɣ	ɣ(100%)
ɪ	e, is, iθ, o, s	e, i, j	e, i, j, l, s, x, ɛ, ɣ, ʝ, θ	i(57%), s(14%), o(14%), e(14%)
ɪː	ɪː	e, f, i, j, u	e, i, ɛ	i(100%)
ɲ	n, gn, ɲ	f, m, n, ŋ, g, ɲ	b, m, n, p, ɲ	ɲ(62%), n(19%), g(19%)
ɾ	i	a, b, d, m, r, s, ŋ, ɔ, g, ɾ	b, r, g	ɾ(89%), i(11%)
ʃ	sθ, θ, θi, θin	s, θ	s	θ(77%), j(15%)
ʊ	o, u	o, u	o, u, w, ɔ	u(71%), ʊ(14%), w(8%)
ʎ	l, x, ʎ	f, l, ʎ	l, x, ʎ	x(73%), l(18%), ʎ(9%)
ʒ	ʃ, β	k, s, x, ɣ	k, p, s, x, g	x(66%), j(15%), ʝ(8%)
ʒː	j, x	a, e, i, j, l, o, s, u, x, θ	e, k, l, p, s, x, ŋ, ɛ, g, ʝ	x(60%), j(40%)

Table D.8: Phones predicted by our different models with a confidence above 0.05, from Italian to Spanish. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

Italian to	Portuguese			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
extipaa	a, ɐ	a, ɐ, ɐ̃, ɑ	a, ɐ, ɐ̃	ɐ(68%), a(15%), ɐ̃(10%)
aː	a	a, i, ɐ, ɐ̃, ɑ	a, ɐ	a(82%), ɐ̃(7%), ɑ(6%)
b	b, v, vɨ	b	b, v	b(80%), v(19%)
bː	b	b, p, v, g	b, v	b(100%)
d	d	d, t	d	d(89%)
dː	d	b, d, t, v	d, n, p, t	d(100%)
e	e, eɪ, eŋ, ɨ	e, i, ɛ, ɨ	e, ɛ, ɨ	ɨ(37%), e(37%), i(9%), ŏ(8%)
eː	e, eɪ, ɛ	e, i, ɛ, ɨ	e, ɛ	e(71%), ɛ(16%), i(10%)
f	f	f	f	f(97%)
i	e, eɪ, i, iŋ, ɐ̃, ɨ	e, i	e, i	i(77%), e(7%), j(7%)
iː	i	e, i, j	i	i(90%)
j	i, l, ɾ, ʃ, ʒ	e, i, j, n, o, s, z, ɛ, ɨ, ɾ	e, i, ɛ, ɨ	i(38%), ʃ(21%), l(17%), ɾ(7%), ʒ(7%), ɾ(7%)
k	k	k	k	k(91%), g(8%)
kː	k	k	k	k(92%), g(8%)
l	l	l	l	l(91%)
m	m	m	m	m(98%)
n	n, ŋ, ɪŋ	n, ɪ	n	ŋ(54%), n(38%)
o	u, ʊ	o, u, ũ, ɔ	o, u, ɔ	ʊ(60%), u(27%)
oː	o, oŋ, ɔ	o, u, ũ, ɔ	o, u, ɔ	o(71%), ŏ(16%), u(5%)
p	p, pɨ	p	p	p(94%)
pː	p	p	p	p(100%)
r	ɹ, ɾ, ʁ	ɾ, ʁ	ɾ, ʁ	ɹ(54%), ɾ(36%), ʁ(8%)
s	s, ʃ	s, ʃ	k, s, z, ʃ	s(58%), ʃ(35%)
sː	ʒ	k, p, s, z, ʃ, ʒ	p, s, z, ʃ, ʒ	s(50%), z(33%), ʒ(17%)
t	d, t, ʃ	t	t	t(77%), d(15%)
tː	t	t	t	t(90%), d(7%)
u	o, ow, u, ũ	o, u, ũ	o, u, ũ, ɔ	u(63%), o(22%), ũ(15%)
uː	ow, u	o, u, ũ	o, u, ũ, ɔ	u(90%), o(6%)
v	v	v	b, v	v(95%)
w	u, w	d, k, n, o, t, u, ũ, v, ɐ̃, g	a, o, u, ɐ̃, ɔ	w(81%), u(19%)
z	s, z, ʃ, ʒ	s, z, ʃ, ʒ	s, z, ʃ, ʒ	z(89%)
zː	j	k, s, g, ʒ	e, i, u, ɛ, ɨ, ʃ	j(100%)
ŋ	ŋ, ɪŋ	k, n, ũ, ŋ, ɐ̃, g, ɪ	n, p	ŋ(100%)
ɔ	u, ɔ	o, u, ũ, ɔ	o, u, ɔ	ɔ(46%), u(38%), o(11%)
ɔː	o, ow, oɪ, ɔ	o, u, ũ, ɔ	o, ɔ	ɐ̃(49%), o(28%), ɔ(17%)
ɛ	e, eɪ, eŋ, ɛ	e, i, ɛ, ɨ	e, ɛ	e(59%), ɛ(22%), ɨ(8%), ɪ(6%)
ɛː	e, eɪ, ɛ, ɨ	e, i, ɛ, ɨ	e, ɛ	ɛ(50%), e(41%), ɨ(6%)
g	g	k, g	g	g(95%)
gː	g	k, g	k, g	g(100%)
ɪ	iʃ, ɐ, ɨ, ɨʃ, ʊ	e, i, ɛ, ɨ	e, i, ɛ, ɨ, ʃ	ɨ(62%), i(8%), ŏ(8%), ʊ(8%), ɪ(8%), ɐ(8%)
ɪː	ɪ	d, f, i, k, s, ʒ	e, i, ɛ, ɨ	ɪ(100%)
ɲ	n, g, gn, ɲ	n, ɪ, ɲ	n, p, ɲ	ɲ(70%), g(15%), n(10%)
ɾ	ʃ	d, s, v, z, g, ɪ, ɾ, ʁ, ʒ	e, i, l, s, ɛ, ɨ, ɹ, ɾ, ʁ	ʁ(71%), ɹ(13%), e(10%), z(6%)
ʃ	eŋʃ, s, ɪʃ, ʃ	s, z, ʒ	e, k, p, s, z, ɛ, ɨ, ʃ	s(61%), ʃ(18%), z(5%)
ʊ	u, ʊ	o, u, ũ	o, u, ũ, ɔ	u(83%), ʊ(13%)
ʎ	ʎ	l, ʎ	l	ʎ(100%)
ʒ	ʒ	ʒ	a, k, p, s, ɛ, g, ɨ, ʃ, ʒ	ʒ(84%), j(6%)
ʒː	ʒː	n, s, v, z, g, ɪ, ʊ, ʒ	b, e, k, o, p, s, ɔ, ɛ, g, ʃ	ʒ(75%), ɪ(25%)

Table D.9: Phones predicted by our different models with a confidence above 0.05, from Italian to Portuguese. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

Italian to	French			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
extipaa	a, al, aʁ, aʁd, aʁde, e	a, ą̃, ɛ	a	a(72%), ą̃(10%), e(7%)
aː	a, aa, aak, aaks, aj, e	a, ɛ	a	a(56%), ɛ(21%), e(7%), ą̃(5%)
b	b	b	b, v	b(98%)
bː	b, v	b, p, v, ʒ	b, v	b(67%), v(33%)
d	d, di	d, g, ʒ	d	d(85%), ʁ(5%)
e	e, ine, ize, ɛ	d, e, f, i, v, ɛ	a, e, i, j, w, ą̃, ə, ɛ	e(51%), ɛ(20%), ą̃(11%), i(6%)
eː	e, i, w, wa, ą̃	d, e, i, ɛ, ɛ̃	a, e, ą̃, ɛ	i(23%), e(23%), ɛ(23%), a(15%), ɛ̃(8%), ą̃(8%)
f	f, fʁ	f	f	f(98%)
i	aj, e, i, ik, j	i, j, ɛ̃	i	i(67%), ɛ̃(12%), j(11%)
iː	i	i, ɛ̃	i	i(80%), ɛ̃(15%)
j	j, l	e, i, j, ɛ, ɛ̃	a, d, e, i, j, w, ą̃, ə, ɛ, ʒ	j(54%), l(38%), ʒ(8%)
k	k	k	k	k(94%)
kː	k	k	k	k(100%)
l	al, l	l	l	l(96%)
m	m, mism	m	m	m(95%)
n	an, n, ni, ɛ̃	n	n	n(91%)
o	e, je, o, u, y, ɔ, ɔn, ɔ̃, ɛ	f, k, y, ø, œ̃, ɔ, ɔ̃	o, y, ɔ	ɔ(48%), ɔ̃(15%), ø(7%), y(7%), u(6%)
oː	kz, u, y, œ, ɔ, ɔ̃	y, ø, œ̃, ɔ, ɔ̃	o, ɔ	œ(31%), ɔ̃(22%), ɔ(19%), o(8%), u(6%)
p	ip, p, pa	k, p	p	p(97%)
pː	p	k, p	p	p(83%), ʃ(17%)
r	ʁ	ʁ	ʁ	ʁ(93%)
s	ks, s	e, s, ʃ	s	s(88%)
sː	z	a, e, k, o, p, s, z, ɛ, ʃ	p, s, z, ʃ	z(100%)
t	t	d, e, s, t, ʃ	t	t(85%)
tː	kt, pt, t	e, k, p, s, t, ʃ	t	t(61%), k(33%), j(6%)
u	y	k, y, œ̃	y, ɔ	y(82%)
uː	y	y, œ̃	y	y(87%), u(13%)
v	de, f, v	d, f, v, ʒ	v	v(64%), f(21%), d(6%)
w	w	e, f, i, œ̃, ɛ̃	a, ɔ	s(100%)
z	s, z, za, gz	p, s, t, z, ʁ	s, z, ʃ, ʒ	z(66%), s(19%), j(13%)
ŋ	ŋ	n	m, n, p, ą̃, ɲ	
ɔ	o, wa, ø, ɔ, ɔb, ə-	y, ø, œ̃, ɔ, ɔ̃	ɔ	ɔ(77%), o(8%), u(5%)
ɔː	i, ɔ, ʁɔ	y, ø, œ̃, ɔ	ɔ	ɔ̃(56%), ɔ(37%)
ɛ	e, ą̃, ɛ	e, i, ə, ɛ	a, e, w, ą̃, ə, ɛ	ɛ(35%), ą̃(27%), e(23%)
ɛː	e, ɛ	e, i, w, ə, ɛ	a, e, w, ą̃, ɛ	ɛ(77%), e(14%)
g	g, ʒ	k, g	g, ʒ	g(83%), k(9%), ʒ(6%)
ɪ	ɪ	i, j, ɛ, ɛ̃	a, d, e, i, j, l, s, v, ɛ, ʒ	i(100%)
ɲ	j, nj	f, n	m, n, ą̃, ɲ	n(43%), ɲ(43%), ʁ(14%)
ɾ	e	a, e, s, v, ɛ, ʁ	ʁ	ʁ(100%)
ʃ	o, s	s, z, ʃ, ʒ	s, ʃ, ʒ	s(74%), j(9%), ʃ(6%), ʁ(6%)
ʊ	y, ɔʁ	y, œ̃	y, ɔ	y(82%), o(10%)
ʎ	j, l	l	l, ʒ	l(67%), j(33%)
ʒ	z	e, j, k, s, ʃ, ʒ	ʒ	ʒ(81%), z(5%), j(5%), ʁ(5%)
ʒː	j	e, i, j, p, s, v, z, ɛ, ʒ	j, v, g, ʃ, ʒ	ʒ(67%), j(33%)

Table D.10: Phones predicted by our different models with a confidence above 0.05, from Italian to French. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

Italian to	Romanian			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
extipaa	a, at, ə	a, ə	a	ə(50%), a(34%), ɨ(7%), e(6%)
aː	a, aɪ	a	a	a(89%)
b	b	b	b	b(94%)
bː	b, o, gu	b, p	b	g(50%), b(50%)
d	ad, d	d, z	d, z	d(90%)
e	ae, e, ea, ja, je, te, ə	a, e, i, j, ʒ	e, i, j, ə	e(72%), ə(9%), a(8%), i(6%)
eː	e, ea, i, ɨ	e, i, j	a, e, i, j, ə	e(67%), i(22%), ɨ(11%)
f	f	f	f	f(100%)
i	e, ea, est, i, in, ine, l, r, ə, ɨ	e, i, j	e, i, j	i(48%), e(25%), ɨ(12%), ə(8%)
iː	e, i, j, l, si, sj	e, i	a, e, i, j	i(72%), e(14%)
j	e, j, l, ɪ	e, i, j	j, ʃ	l(33%), n(20%), j(20%), s(7%), p(7%), ɔ(7%), ɪ(7%)
k	k	k	k	k(95%)
kː	k, p	k	k	k(83%), t(17%)
l	a, l, la, lə	l	l, r	l(70%), ɾ(13%), r(6%)
m	am, m	m	m	m(98%)
n	n	n	n	n(87%), ŋ(6%)
o	ar, o, u, ɔa, ə, ʊ	a, o, u, ɔ	a, j, o, ʊ	u(55%), e(11%), ʊ(8%), o(7%)
oː	o, p, u, ɔa	a, o, u, ɔ	o, u, ɔ, ʊ	u(37%), o(37%), ɔ(11%)
p	p, pj	p	p	p(100%)
pː	p	p	p	p(100%)
r	ar, r, ra	r	r	r(72%), ɾ(16%)
s	as, s, ʃ	s, ʃ	s, ʃ	s(83%), ʃ(14%)
sː	s	s, ʃ	s	s(100%)
t	st, t, ts	t	t	t(96%)
tː	apt, bts, pt, t	k, p, s, t, ʒ	k, p, t	t(63%), p(32%), b(5%)
u	u, ɨ	a, o, u	u, ɔ	u(70%), ɨ(10%)
uː	a, t, u, wa	a, o, u	u, ɔ	u(77%)
v	b, v	h, v, ʒ	j, v	v(73%), b(12%), j(6%)
w	w	a, o	a, j, k, o, ʊ	p(50%), b(50%)
z	s, ʃ	s, ʃ	s	s(85%), ʃ(10%)
zː	zː	e, i, j, s, ʃ, ʒ	a, d, e, i, j, s, ə, ɨ, ʃ, ʒ	z(100%)
ŋ	m, ŋ	a, i, u, ɨ	m, n, ŋ	ŋ(67%), m(33%)
ɔ	o, u, ɔa	a, o, ɔ	o, ɔ	u(51%), ɔ(27%), o(22%)
ɔː	a, au, aʊ, o, u, ɔa	a, o, ɔ	a, j, k, o, r, ʊ	o(55%), u(24%), ɔ(12%)
ɛ	e, ea, je, jes, jest, jeste, se	a, e, i, j, ɨ	a, e, j, ə, ɨ	e(68%), i(13%), ɨ(8%), a(5%)
ɛː	a, e, i, ja, je, ə	e, j	a, e, j	e(64%), a(16%), i(12%), ə(8%)
g	ag, k, g	d, k, g, ʒ	k, g	g(72%), k(25%)
ɪ	dʒe, jj	e, i, j	e, i, j	j(25%), i(25%), e(25%), d(25%)
ɲ	m, mn	f, i, m, n, ɨ	m, n, p	m(60%), n(20%), j(20%)
ɾ	aɾ, ɾ, ʃ	o, r, s, ʃ	r	n(60%), ɾ(30%), ʃ(10%)
ʃ	aʃ, stʃ, tʃ, ʃ, ʃt	s, ʃ	s, ʃ	ʃ(95%)
ʊ	au, u	a, o, u	o, u, ɔ	u(94%)
ʎ	ie, j, l	f, j, v	l	l(40%), j(40%), ɾ(20%)
ʒ	b, s, ʒ	s, ʃ, ʒ	e, s, ʃ, ʒ	ʒ(93%)
ʒː	z, ʒ	d, e, j, s, v, ʃ, ʒ	e, j, o, r, s, ʃ	ʒ(71%), z(29%)

Table D.11: Phones predicted by our different models with a confidence above 0.05, from Italian to Romanian. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

D.5.3 From Portuguese

Portuguese to	Spanish			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, al, ka, la, na	a	a	a(99%)
b	al, b, bl, eβ, leβ, β	b, p	b, p, β	β(66%), b(32%)
d	d	b, d	d	ð(56%), d(42%)
e	e, ɛ	a, b, d, e, i, j, ɛ, j̱	e, ɛ	e(48%), ɛ(43%)
f	f	f	f	f(98%)
i	e, i, il, in	i	i	i(86%), j(7%)
j	i, j	i	e, i, j, ɛ, j̱	j(94%)
k	k	k	k	k(98%)
l	el, l	l, ʎ	l, ʎ	l(94%), ʎ(5%)
m	m	m	m	m(98%)
n	n	m, n, ɲ	n	n(97%)
o	lo, o, we, ɔ, ɔr	o, w, ɔ	o, u, w, ɔ	ɔ(51%), o(35%)
p	p	p	p	p(99%)
r	ɾ	r, ɾ	a, i, l, m, n, r, u, ɔ, ɛ, ɾ	ɾ(100%)
s	s, sja, θ	s, θ	s, θ	s(53%), θ(46%)
t	t	t	t	t(99%)
u	o, u, ulo, un, ɔ, ɔn	o, u, ɔ	o, u, ɔ	u(42%), o(30%), ɔ(24%)
ũ	u	o, u, ɔ	o, u, ɔ	u(92%), ɔ(8%)
v	b, be, ben, eβ, β	b, f	b	β(66%), b(32%)
w	l, w	a, b, f, l, u, w	a, f, k, o, t, u, w, ɔ	l(48%), w(38%)
z	ks, s, θ	s, θ	s, θ	s(85%), θ(13%)
ŋ	n, na, naɾ	m, n, ɛ	m, n, ŋ, ɲ	n(85%), m(8%), ŋ(5%)
ɐ	a, al, ana, la, na	a	a	a(98%)
ɐ̃	a, ana	a, b	a	a(58%), j̱(34%), ɔ(7%)
ɑ	a, na	a	a	a(100%)
ɔ	o, we, wo, ɔ	o, w, ɔ	o, w, ɔ	ɔ(70%), o(19%), e(5%)
ə	b, d, ɛ	p, r, ɾ	a, b, e, f, l, p, r, ɛ, ɾ	θ(50%), e(25%), ɛ(25%)
ɛ	e, ek, inje, je, nje, ɛ	a, b, e, i, j, ɛ, j̱	e, i, j, ɛ	ɛ(49%), e(47%)
ɡ	g, ɣ	k, g, ɣ	k, g, ɣ	ɣ(68%), g(28%)
ɨ	e, ɛ	e, i, j, ɛ, j̱	e, ɛ	e(71%), ɛ(25%)
ɪ	a, e, i, n, nj, ɪ, ʃ, θ	a, e, f, i, j, l, m, s, x, j̱	e, i, j, l, s, x, ɛ, g, ɣ, j̱	ɛ(37%), ɪ(12%), n(10%), i(8%), o(6%), j(6%), e(6%), θ(6%)
ɲ	eɲ, n, ɲ	a, e, f, i, j, m, n, ɛ, ɣ, ɲ	n, ɲ	ɲ(56%), n(36%)
ɹ	ɛɾ, ɾ	r, ɛ, ɾ	r, ɾ	ɾ(99%)
ɾ	ɾ	r, ɾ	r, ɾ	ɾ(99%)
ʁ	r, re	r	r, ɾ	r(99%)
ʃ	es, s	k, l, p, s, x, ʎ, θ	s	s(85%), θ(8%)
ʊ	jo, lo, o	f, i, o, u, w, ɔ	o, u, ɔ	o(89%)
ʊ̃	no, ɔn	f, i, j, o, u, ɔ	a, e, i, n, o, u, ɔ, ɛ, β, θ	ɔ(71%), n(19%), o(7%)
ʎ	l, x, ʎ	f, i, l	l, ʎ	x(50%), l(33%), ʎ(17%)
ʒ	s, x	f, i, j, x, j̱	k, s, x, g, j, θ	x(80%), s(8%), j(7%)

Table D.12: Phones predicted by our different models with a confidence above 0.05, from Portuguese to Spanish. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

Portuguese to	Italian			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, aː	a, aː	a, aː	aː(66%), a(33%)
b	b, p	b, p	b, p	b(73%), p(13%)
d	d, di, t	d, v	d	d(71%), t(26%)
e	a, de, e, eː, le, ne	e, eː, i, iː, v, ɛ, ɛː	a, e, eː, i, ɛ, ɛː	e(53%), ɛ(21%), i(7%)
f	f, ff	f	f	f(98%)
i	e, i, in, iː	i, iː	i, iː	i(64%), iː(21%), e(9%)
j	i	e, i, iː, ɪ	aː, d, e, eː, i, iː, j, v, ɛ, ɛː	i(72%), s(12%), ʒ(9%), ʃ(5%)
k	k, kw	k	k	k(95%)
l	l, le, llo	l	l	l(97%)
m	m, mm	m	m	m(99%)
n	n, nn	m, n, ŋ, ɲ	m, n	n(98%)
o	alo, o, oː, ɔ, ɔː	o, oː, u, uː, w, ɔ, ɔː, ʊ	o, oː, u, ɔ, ɔː, ʊ	oː(36%), o(22%), ɔː(19%), ɔ(12%), u(7%)
p	p, pr	p	p	p(92%), pː(7%)
r	r	r	e, l, o, r	r(100%)
s	s, ss, tʃ	s	s	s(69%), ʃ(19%), t(5%)
t	t	t, tː	t	t(88%), tː(11%)
u	o	o, u, uː, ɔ, ɔː, ʊ	o, u, uː, ɔ, ʊ	o(50%), ʊ(21%), uː(10%), ɔ(8%), u(6%)
ũ	o, u	o, u, uː, ʊ	o, u, uː, ɔ, ʊ	o(61%), u(21%), ʊ(14%)
v	v, ve, ven	f, v	b, v	v(87%), b(7%)
w	d, di, l, w	f, l, o, w, ɾ, ʊ	a, aː, k, o, oː, u, uː, ɔ, ɔː, ʊ	w(59%), l(29%), d(6%), ʊ(6%)
z	d, dz, s, tʃ, tʃi, z, ɔː	d, i, l, n, r, s, t, z, ʃ	d, s, t, z	z(78%), ʃ(6%)
ŋ	n, ne	m, n	a, m, n, ŋ, ɲ	n(93%)
ɐ	a, la, na	a, aː	a	a(97%)
ɐ̃	a, aː, va	a, aː, i	a, aː	a(50%), ɔː(28%), aː(12%), i(7%)
ɑ	a, aː	a, aː	a, aː	a(50%), aː(50%)
ɔ	o, ɔ, ɔː, ɔːdo, ɔːla	f, o, oː, uː, w, ɔ, ɔː, ʊ	o, oː, w, ɔ, ɔː, ʊ	ɔ(45%), o(26%), ɔː(22%)
ə	d, t	r	a, aː, e, l, o, r, s, v, ɛ, ʃ	t(50%), d(25%), o(6%), iː(6%), e(6%), aː(6%)
ɛ	anɛ, e, eː, inɛ, nɛ, ɛ, ɛː, ɛːtro	aː, e, eː, i, iː, j, v, ɛ, ɛː	a, aː, e, eː, i, j, ɛ, ɛː	ɛ(45%), ɛː(32%), e(12%), eː(7%)
g	k, g, gw	k, g	k, g	g(74%), k(20%)
i	e, ine, le, ɪ	d, e, eː, f, i, v	a, e, eː, i, o, ɛ, ɛː	e(80%), i(7%)
ɪ	a, d, dʒe, e, i, s, ve, ɪ	a, b, d, e, i, o, s, v, ɛ	a, d, e, i, iː, r, s, ɛː, ʃ, ʒ	n(20%), ɛ(18%), e(16%), t(7%), i(7%), d(7%), a(5%)
ɲ	n, ne, ɲ, ɲj	f, i, l, m, n, v, ɲ	a, m, n, ŋ, ɲ	ɲ(59%), n(36%)
ɾ	dere, nere, re	r	r	r(98%)
ɾ	r	r	r	r(98%)
ʁ	r, rr	r	r	r(73%), ɾ(24%)
ʃ	s, tʃe	k, p, s, t	e, o, s	s(81%), ʃ(12%)
ʊ	lo, no, o, olo, ro	f, o, u, uː, w, ɔ, ɔː, ʊ	o, oː, u, uː, ɔ, ɔː, ʊ	o(94%)
ʊ̃	ne, no, oːne, ɔːne	f, n, v	b, n, o, v, ʊ	e(63%), o(20%), oː(14%)
ʎ	li, ʎ	i, l	l	ʎ(86%), l(14%)
ʒ	dʒ	d, f, s, v	d, s	ʒ(76%), z(8%)

Table D.13: Phones predicted by our different models with a confidence above 0.05, from Portuguese to Italian. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

Portuguese to	French			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, at, e	a, o	a	a(51%), e(33%), ɛ(9%)
b	b	b, p	b, v	b(94%)
d	d, t	d	d	d(81%), t(15%)
e	e, i, l, ɛ, ɛk	d, e, i, j, l, v, ə, ɛ	e, i, j, ɛ	e(30%), a̋(29%), ɛ(24%), i(6%)
f	f, fi	f	f	f(100%)
i	e, i, ik	i	e, i	i(74%), ɛ̋(10%), e(7%)
j	i, j, n	d, e, f, i, j, ɛ̋, ʒ	a, e, i, j, ɛ	ʁ(60%), j(40%)
k	k, kɔ̃	k	k, ʃ	k(97%)
l	l	l	l	l(99%)
m	m, ma	m	m	m(98%)
n	n	n	n	n(92%)
o	p, y, œ, ɔ	w, y, ɔ	ɔ	œ(23%), ɔ(23%), y(13%), o(13%), ɔ̃(10%), u(8%)
p	p, pa	p	p	p(98%)
s	ks, s	s	s	s(96%)
t	kt, t	t	t	t(96%)
u	y, yk, ɔ	y	u, y, ɔ	ɔ(41%), y(40%), ɔ̃(9%)
ũ	w, y, ɔ	y, őe	u, y, ɔ	ɔ̃(45%), y(18%), őe(9%), ɛ̋(9%), a̋(9%), ɔ(9%)
v	b, de, f, v	b, f, v	v	v(60%), f(21%), b(12%)
w	l, y, ɛ	f, l, u, y	a, f, i, k, o, y, ɔ	l(88%), t(12%)
z	s, z, gz	p, s, z, ʁ	s, z, ʒ	z(81%), j(8%), s(6%)
ŋ	m, n, a̋	m, n	n, a̋, ɲ	n(30%), k(20%), g(20%), p(10%), m(10%), ʁ(10%)
ɐ	a, ak, al	a	a	a(88%)
ɐ̃	j, a̋	a, b, j, o, a̋, ɛ	a	j(54%), a̋(38%)
ɑ	a, nɛ, ɛ	a	a	a(67%), ɛ(22%), o(11%)
ɔ	o, wa, ɔ	ɔ	ɔ	ɔ(89%)
ə	ə	ʁ	d, e, j, l, n, s, t, v, ʁ, ʒ	a̋(50%), ɛ(50%)
ɛ	e, nɛ, ɛ, ɛk	e, i	e, i, j, ɛ	ɛ(65%), e(30%)
g	k, g, ʒ	f, k, g	g, ʒ	g(67%), k(13%), n(8%), j(5%), ʒ(5%)
i	e, jasjɔ̃, ɛ, ɛk, ɛʁ	e, i, ɛ	e, ɔ, ɛ	e(44%), ɛ(36%), i(9%)
ɪ	k, p, ʒ	f, i, j, k, m, s, y, ɛ, ɛ̋	a, d, e, i, j, v, ɛ, ʃ, ʒ	k(40%), y(10%), p(10%), i(10%), a(10%), a̋(10%), ʒ(10%)
ɲ	n	f, n, ɲ	n, a̋, ɲ	n(60%), ɲ(40%)
ɹ	tœʁ, ʁ, ʁa, ʁal, ʁatif	ʁ	ʁ	ʁ(89%)
ɾ	ʁ	ʁ	ʁ	ʁ(95%)
ʁ	ʁ	ʁ	ʁ	ʁ(100%)
ʃ	s	k, p, s, ʃ	s, ʃ	s(81%), e(5%)
ʊ	aʒ, e, i, k, l, o, s, ɛ, ɛʁ, ɛ̋	f, k, m, y, ɔ	k, o, y, ɔ	ø(20%), l(18%), y(10%), o(10%), ɛ(10%), ɔ̃(8%), i(7%), a(7%)
ɔ̃	ɔ̃	f, iː, n, őe	b, f, i, j, n, s, v, y, ɔ	ɔ̃(98%)
ʎ	j, jl, l, lj	l	l	l(71%), j(29%)
ʒ	ʒ	ʒ	ʒ	ʒ(91%), s(6%)

Table D.14: Phones predicted by our different models with a confidence above 0.05, from Portuguese to French. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

Portuguese to	Romanian			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a	a	a	a(93%)
b	b, p, pə	b	b	b(71%), p(24%)
d	d, t, z	d, z	d, z	t(49%), d(42%), z(8%)
e	a, e, ea, je, jede, ə	d, e, i, j, l, s, v, ʃ, ʊ, ʒ	a, e, i, j, ə, ɨ	e(63%), i(24%), a(7%), ə(5%)
f	f	f	f	f(100%)
i	e, eg, i, j, je, sj, ə	j	a, e, i, j	i(54%), e(21%), ə(10%), j(7%)
j	j	j, v, ʃ	a, e, j, v, ʒ	j(60%), i(20%), ɪ(20%)
k	k	k	k	k(95%)
l	l	l	l	l(82%), ɾ(8%)
m	m	m	m	m(93%)
n	n	m, n, ɨ	n	n(100%)
o	a, o, u, ul, ə	a, o, ɔ	a, o, ɔ	u(51%), o(30%), ɔ(9%)
p	p	p	p	p(100%)
s	s, sə, səd, tʃ, ʃ	s	s	s(67%), ʃ(33%)
t	pt, st, t, ts	p, s, t	t	t(99%)
u	juo, o, u, ud, un, uo	a, d, j, u	a, j, u, w, ɔ, ə	u(79%), o(12%)
ũ	u	u	u	u(100%)
v	b, v, ve	v	v	v(66%), b(17%)
w	a, d, de, l, ʊ	a, d, f, j, l, z, g, ʒ	a, j, ɔ	t(25%), p(25%), l(25%), d(25%)
z	s, ts, tʃ	s, ʃ	s, ʃ	s(55%), ʃ(36%), r(9%)
ŋ	n, ne	m, n, ɨ	n	n(80%), m(7%), ŋ(7%)
ɐ	a, nə, rə, ə	a	a	ə(66%), a(17%), e(13%)
ɐ̃	a, an, ɨ, ɨnə	a	a	a(40%), ɨ(37%), u(10%), ɣ(7%), ə(7%)
ɑ	a, ə	a	a	a(80%), ə(20%)
ɔ	aɪ, o, od, u, ɔa	o, ɔ	a, o, ɔ	ɔ(44%), o(31%), u(19%), ɪ(6%)
ə	k	r	a, j	k(33%), d(33%), ə(33%)
ɛ	dʒe, e, ea, je, se, sea	e, j, v	a, e, j	e(95%)
ɡ	k, g	d, k, g, ʒ	g	k(48%), g(48%)
ɨ	e, ə	d, e, j, s, v, ʃ, ʒ	a, e, j	e(82%), ə(9%)
ɪ	d, de, dʒe, n, p	a, d, f, h, j, l, o, s, v, ʒ	e, i, j, ə, ɨ	e(29%), ɨ(21%), u(8%), p(8%), a(8%)
ɲ	ɲ	f, i, m, n, ɨ	m, n, ɨ	n(100%)
ɹ	a, are, de, dea, j, ne, r	r	a, r	r(60%), j(12%), ɾ(9%), d(5%)
ɾ	n, r, ɪ, ɾ	r	r	r(69%), ɾ(26%)
ʁ	r	r	r	r(71%), ɾ(19%), j(10%)
ʃ	as, s, st, tʃe	s	s	s(73%), ʃ(16%)
ʊ	a, d, in, n, r, u, v, ə, g, ʊ	a, d, f, j, o, s, u, v, ʒ	a, j, o, u, ɔ, ə, ʊ	e(21%), ʊ(21%), u(15%), r(9%), n(9%), ə(9%), a(6%)
ʊ̃	n, ne, nə, me	f, n, v, ɨ	a, i, n, t, u, v, y, ɨ, ʊ	n(38%), e(25%), ʊ(12%), ɪ(12%), ə(12%)
ʎ	ɪ	e, i, j, l, ʒ	l	l(50%), ɪ(50%)
ʒ	dʒ, s, st, ʒ	h, ʒ	d, s, v, z, ʃ, ʒ	ʒ(81%), s(14%)

Table D.15: Phones predicted by our different models with a confidence above 0.05, from Portuguese to Romanian. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

D.5.4 From French

French to	Spanish			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, aðo	a	a	a(94%)
aː	a	a	a	a(100%)
b	b, bo, βjo, βo	b, β	a, b, f, k, β	β(67%), b(30%)
d	d, do, jo, ðjo, ðo	d	d	ð(53%), d(43%)
e	ad, aðo, aɾ, e	a, b, d, e, i, j, o, ɛ	b, e, k, p, s, ɛ, ʎ, j, θ	e(42%), a(33%), ɛ(18%)
f	f, o, βo	f, i, u	f, o, u, ɔ	f(77%), β(20%)
i	i, ia, io, ito, iðo, ja	i	i	i(92%)
j	j, jo, lo	a, e, i, j, l, o, u, w, ɔ, ɛ	a, e, i, j, l, s, ɛ, ɪ, j, θ	j(85%)
k	k, ko	k	k	k(92%)
l	l, la, le, ljo, lo	l, ʎ	l	l(93%)
m	m, ma, me, mo	m	m	m(98%)
n	n, na, no	n	b, m, n	n(93%)
o	al, aʊ, o, wo, ɔ	a, o, u, ɔ	k, o, w, ɔ	o(48%), ʊ(28%), ɔ(16%), a(8%)
p	p, pa	k, p, t, ʎ	p	p(94%)
s	es, s, se, so, θ, θjo	k, l, s, t, θ	k, s, θ	s(53%), θ(41%)
t	t, ta, te, tjo, to	t	t	t(83%), ð(13%)
u	eʎo, o, oβo	o, u, ɔ	o, u, ɔ	o(64%), ɔ(21%), u(14%)
v	b, β, βe, βo	b	b	β(58%), b(40%)
w	e, ɔ, ɾe	a, e, i, n, o, u, ɔ, ɛ, β	o, u, ɔ	ɾ(40%), w(20%), t(20%), θ(20%)
y	o, u, uðo, we, wo	o, u, ɔ	o, u, ɔ	u(79%), o(10%)
z	s, sis	s, θ	s	s(83%), θ(10%)
ð	ta, to	k, t	d, t	t(50%), o(50%)
ø	oso	a, o, u, ɔ	o, w, ɔ	o(83%), e(17%)
œ	aðɔ, ɔ	o, u, ɔ	o, ɔ	ɔ(82%), o(9%), a(9%)
œ̃	une, uno, unos	o, u, ɔ	u, ɔ	o(50%), e(25%), ɛ(25%)
ɑ̃	an, ano, ante, jɛnto, ɛnte, ɛnto	a, ɛ	a, ɛ	ɛ(58%), a(40%)
ɔ	o, ɔ, ɔn	o, u, ɔ	o, w, ɔ	ɔ(51%), o(45%)
ɔ̃	ɔn	k, n, o, u, ɔ	o, u, ɔ	ɔ(86%), u(7%)
ə	a, e	a, e, f, i, j, l, o, s, u, β	a, b, e, ɛ	e(90%), a(10%)
ɛ	a, ajja, e, es, eso, eto, o, ɛ	a, e, i, p, ɛ, θ	e, ɛ	e(44%), ɛ(36%), a(14%)
ɛː	ɛː	a	a, b, e, ɛ, j	a(100%)
ɛ̃	ano, in, ino	i	a, i, m, ɛ, ʃ	i(89%)
g	g, ɣ, ɣa, ɣo	k, g, ɣ	k, g, ɣ, ʎ	ɣ(58%), g(31%)
ɪ	i	e, i, u	a, e, i, j, l, p, s, ʃ, j, θ	i(100%)
ɲ	ɲja, ɲa	n, ɛ, ɲ	b, m, n, p	n(50%), ɲ(50%)
ʁ	aɾ, paɾ, r, ra, ɾ, ɾa, ɾe, ɾjo, ɾo, ɾɔ	a, d, r, ɛ, ɾ	r, ɾ	ɾ(79%), r(9%)
ʃ	esk, k, pjo	i, k, t, ʃ, j	k	k(50%), s(25%), j(12%), ʃ(12%)
ʒ	x, xe, xjo, ɣo	f, j, x, g, j	e, k, l, x, g, ɣ	x(69%), j(10%), ɣ(7%)
θ	t	t	t, θ	t(100%)

Table D.16: Phones predicted by our different models with a confidence above 0.05, from French to Spanish. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

French to	Italian			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a	a, aː, o	a, aː	a(79%), aː(14%)
aː	a	a, aː	a, aː	a(50%), aː(50%)
b	b, bo	b, v	a, aː, b, f, k, kː, v, g	b(87%), s(6%)
d	da, de, dia, dine, dio, do	d	d, f	d(95%)
e	a, aːre, e, ere, iːre, kaːre, gaːre	aː, e, eː, i, iː, j, oː, v, ɛ, ɛː	a, aː, e, eː, i, j, o, p, s, t	e(64%), a(10%), ɛ(9%), i(7%)
f	f, vo	f	f	f(76%), v(20%)
i	e, i, ia, iːa, o	aː, e, eː, i, iː, j, ɛ, ɛː	i, iː	i(73%), iː(15%)
j	i, le, llo	a, e, i, o, u, ɛ, ʊ	a, aː, d, i, iː, k, s, t, tː, w	i(49%), j(10%), z(9%), r(9%), ʃ(7%)
k	ko	a, d, k, o, s, t, tː, g	k	k(81%), tː(5%)
l	ile, la, le, lle, lo	aː, l, o, s, v, ɛ, ɾ	l	l(93%)
m	m, ma, me, mento, mo	m	m	m(98%)
n	n, na, ne, no	n, ŋ	a, b, m, n, ɲ	n(92%)
o	aʊ, lda, o, passe, ɔ, ɔːko	a, aː, o, oː, u, uː, ɔ, ɔː, ʊ	a, aː, k, o, oː, ɔ, ɔː	o(23%), ʊ(23%), a(13%), ɔ(13%), oː(10%), e(7%)
p	p, po	p	p	p(91%)
s	s, sa, sso, tʃe, tʃo	s, t, z	s, t	s(73%), ʃ(17%)
t	ta, te, to	t, tː	t	t(88%), tː(5%)
u	o, ollo, uːpo	o, oː, u, uː, ɔ, ɔː, ʊ	o, oː, u, uː, ɔ, ʊ	o(57%), uː(14%), ɔ(14%), oː(10%)
v	v, ve, vo	v	a, aː, b, bː, f, v, w, ɛː	v(92%)
w	e, eː, u	aː, e, eː, i, iː, j, o, oː, ɛ, ɛː	a, aː, e, o, oː, u, w, ɔ, ɔː, ʊ	t(25%), l(25%), d(25%), ʒ(25%)
y	o, u, usto, uːe, uːto, ʊ, ʊo	u, uː, ʊ	o, u, uː, ʊ	ʊ(48%), u(22%), o(11%), uː(10%)
z	dʒo, z	s, sː, z, ʃ, ʃː	d, s, t, z	z(79%), ʒ(8%), s(5%)
ð	to	t, tː	a, d, e, f, i, m, o, t	t(100%)
ø	ko, oːzo	o, oː, sː, u, uː, ɔ, ɔː	e, o, oː, s, ɔ	o(47%), oː(37%), ɛ(11%), ɔ(5%)
œ	atoː, oː, toː, v	o, oː, u, uː, ɔ, ɔː, ʊ	o, oː, ɔ	oː(64%), o(7%), e(7%), a(7%), ʊ(7%), ɔː(7%)
œ̃	uːne, uːno	o, u	u	u(33%), o(33%), e(33%)
ɐ	a	a, aː, e, o, oː, ɔ	a, aː, o	a(100%)
ɑ̃	a, ante, ente, ento, ne, ɛnte, ɛnto	a, aː, ɛ	a	a(35%), e(29%), ɛ(29%)
ɔ	o	o, oː, u, uː, ɔ, ɔː, ʊ	o, oː, ɔ, ɔː, ʊ	o(62%), ɔ(19%), ɔː(10%), oː(5%)
ɔ̃	o, onko, onno, onte, oːne, ɔːne	n, o, oː, u	a, o, u, ɔ	o(41%), ɔː(38%), oː(14%)
ə	a, e, ɛ	e, eː, i, iː, o, oː, v, ɔ, ɛ, ɛː	a, aː, e, j, l, o, r, s, v, ɛ	e(53%), i(13%), a(13%), ɛ(13%), oː(7%)
ɛ	a, e, etːo, ɛ, ɛtːo	a, aː, e, eː, i, o, ɛ, ɛː	a, aː, e, eː, i, j, o, s, ɛ, ɛː	e(38%), ɛ(22%), ɛː(12%), a(11%), aː(8%)
ɛː	aː	a, aː	a, aː, e, eː, i, j, s, w, ɛ, ɛː	aː(100%)
ɛ̃	aːno, im, in, iːne, iːno	a, aː, d, e, eː, i, n, o, ɛ, ɲ	a, i, iː	i(74%), iː(9%), e(5%)
g	g, ga, go	k, ŋ, g	f, k, g	g(86%), t(5%), k(5%)
ɪ	i	e, i, iː, u, ɪ	a, aː, d, i, k, p, s, t, tː, g	i(100%)
ɲ	nia, ɲja	n, ɲ	a, aː, b, m, n, p, ɲ	n(50%), ɲ(50%)
ʁ	dere, ere, r, ra, re, ro	r, g	r	r(90%)
ʃ	ko	d, k, s, t, tː, g	a, aː, d, k, n, o, oː, p, s, t	k(50%), s(10%), pː(10%), n(10%), g(10%), ʃ(10%)
ʒ	dʒ, dʒo	d, v, g	d, f, k, v, g	ʒ(70%), ʒː(11%), r(7%)
θ	t	t	e, s, t	t(100%)

Table D.17: Phones predicted by our different models with a confidence above 0.05, from French to Italian. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

French to	Portuguese			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, aʊ, ɐ	a, ɐ, ɐ̃, ʊ	a, ɐ	ɐ(66%), a(25%)
aː	a, ɐ	a, ɐ, ɐ̃	a, ɐ, ɐ̃	a(67%), ɐ(33%)
b	b, bjʊ, bʊ	b, v	b, p	b(79%), v(17%)
d	d, djɐ, dɐ, dɨ, dʊ	d	d	d(95%)
e	adɨ, adʊ, aɪ, eɪ, iaɪ, iɪ, ɨ	a, d, e, i, l, s, v, ɐ, ɛ, ɨ	e, i, k, p, s, v, ɛ, g, ɨ, ʃ	a(25%), e(24%), ɨ(21%), i(15%), ɛ(9%)
f	f, vʊ	f	f	f(74%), v(23%)
i	i, idʊ, itʊ, iɐ	e, i, u, ɛ, ɨ	i	i(88%)
j	i, iɐ̃, jʊ, lʊ, ɐ̃, ʎʊ	e, i, n, o, s, t, u, ũ, ɐ̃, ɪ	a, e, i, j, ɛ, g, ɨ, ʒ	ɐ̃(53%), i(24%), j(7%)
k	k, kɐ, kʊ	k	k	k(88%)
l	eʊ, l, lɐ, lʊ	l	l	l(82%)
m	m, mɐ, mɨ, mʊ	m	m	m(95%)
n	n, naɪ, nɐ, nʊ	n	n, p	n(82%)
o	aʊ, al, ʊ	a, o, u, ɐ, ɐ̃, ɑ, ɔ, ʊ	a, o, ɔ	u(30%), ʊ(30%), o(22%), ɔ(13%)
p	p, pjʊ, pɨ, pʊ	p, ʃ	p	p(93%)
s	s, siʊ, sjɐ, sɐ, sʊ, ʃ	s, z, ʃ, ʒ	s, z, ʃ	s(76%), ʃ(21%)
t	t, tɐ, tɨ, tʊ	t	t	t(89%), d(9%)
u	obʊ, u, ɔ	o, u, ũ, ɔ, ʊ	o, u, ũ, ɔ	u(60%), o(30%), ɔ(10%)
v	v, vɐ, vɨ	v	v	v(96%)
w	ũ, ɔ, ɹe	a, e, i, o, u, v, ɔ, ɛ, ɨ, ɪ	o, u, ũ, ɔ	v(20%), e(20%), d(20%), ɹ(20%), ɾ(20%)
y	u, udʊ, ugɐ, uɨ, uʊ	o, u, ũ	u, ũ, ɔ	u(82%), ʊ(6%)
z	z, zjɐ	b, k, o, s, u, z, g, ʃ, ʊ, ʒ	z, ʒ	z(86%), t(6%)
ø	eʊ, ozʊ	o, u, ɔ, ʊ	k, o, p, ɔ	ʊ(73%), o(9%), e(9%), ɔ(9%)
œ	do, o, ɐdo	o, u, ũ, ɔ	o, ɔ	o(91%), ɐ(9%)
œ̃	unɨ, unʊ, ũŋʃ	o, u, ũ	o, u, ũ	ũ(25%), o(25%), ʊ(25%), ɨ(25%)
ɑ̃	eɪŋtɨ, eɪŋtʊ, ɐ̃ŋtɨ	a, e, ɐ, ɐ̃, ɑ, ɛ	ɐ̃	e(50%), ɐ̃(41%)
ɔ	o, u, ɔ	o, u, ũ, ɔ, ʊ	o, ɔ	u(57%), ɔ(33%), o(7%)
ɔ̃	oŋkʊ, ũŋ, ũŋdɐ, ũŋkʊ, ʁaɪʊ, ő	o, u, ũ	o, u, ũ, ɐ̃	ő(55%), u(20%), ũ(9%), o(8%)
ə	e, ɐ, ɨ	a, d, e, i, l, s, v, z, ɛ, ɨ	a, b, e, i, v, ɐ, ɛ, g, ɨ	e(56%), i(12%), ɨ(12%), ɔ(6%), ɛ(6%), ɐ(6%)
ɛ	a, e, eɪ, ɐ, ɛ, ɨ, ʊ	a, e, i, ɐ, ɛ, ɨ	e, i, ɛ, ɨ	e(29%), ɨ(29%), ɛ(20%), a(9%)
ɛː	ɛː	a, ɐ, ɐ̃, ʊ	a, e, ɛ, ɨ	a(100%)
ɛ̃	inʊ, iŋ	e, i, o, u, ɐ̃, ɪ	e, i, ɨ	i(77%), e(10%)
g	g, gɐ	g	g	g(92%), ŋ(8%)
ɪ	i	i	e, i, j, p, ɐ, ɛ, g, ɨ, ʒ	i(75%), ɐ(25%)
ɲ	ɲɐ	n, ɪ, ɲ	b, e, n, p, ɐ, ɐ̃, ɲ	ɲ(100%)
ʁ	ɹ, ɹɨ, ɹʊ, ɾjʊ, ɾɐ, ɾəsʊ, ɾʊ, ʁ, ʁɐ, ʁɨ	ɹ, ɾ, ʁ	ɾ, ʁ	ɹ(40%), ɾ(40%), ʁ(14%)
ʃ	a, k, ʃ	k, ʃ	k, p, s, ʃ	k(71%), ʃ(14%), ɾ(14%)
ʒ	gʊ, ʒ, ʒjʊ, ʒɨ	k, g, ʒ	g, ʒ	ʒ(78%), g(8%)
θ	t	t	s, t	t(100%)

Table D.18: Phones predicted by our different models with a confidence above 0.05, from French to Portuguese. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

French to	Romanian			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, ə	a	a	a(57%), ə(17%), e(13%)
b	b	b, v	b	b(100%)
d	d, də	d, v	d	d(90%), n(10%)
e	a, e, ea, je, s, vale	a, d, e, j, s, ʃ	a, e, j, k, o, s, v, ʃ	e(63%), a(21%), s(5%), o(5%), j(5%)
f	f	f, v	f	f(88%), v(12%)
i	e, i, j, ɨs	e, i, j, ʃ	e, i, j, ə, ɨ	i(52%), e(26%), j(9%), ə(9%)
j	j, o	a, e, i, j, s, u, ə, ʃ	a, e, i, j, o, s, u, ɔ, ɨ, ʃ	s(50%), j(25%), ʃ(25%)
k	k	d, k, r, t, u	k	k(87%), t(7%), j(7%)
l	ale, l, lə, r, rə	a, e, i, j, l, u, ʒ	l	l(69%), r(10%), j(7%), ɾ(7%)
m	m, mə, əmə	m	m	m(95%), n(5%)
n	n, nə	n	m, n	n(80%), m(10%), ɾ(10%)
o	a, al, o, oʊ, pasəre	a, o	a, k, o	o(33%), a(33%), u(17%), ə(17%)
p	p	p	p	p(89%), t(11%)
s	as, s, sət, tʃ, tʃe, ʃ	s, ʃ	s	s(50%), ʃ(39%), t(6%), r(6%)
t	st, t, tə	t	t	t(87%), n(9%)
u	o, oj, u, ul, up	o, u	a, o, u, ɔ	u(80%), o(20%)
v	b, bə, v	v	b, f, k, v	v(69%), b(19%), s(6%), j(6%)
w	e	a, d, e, j, o, s, v, y, ʊ	a, j, k, o, s, u, ɔ, ɨ, ʊ	e(40%), w(20%), j(20%), ʒ(20%)
y	oj, u, ur, ut	o, u	o, u, ɔ	u(83%), o(8%), e(8%)
z	s	j, s, ʃ	l, r, s, t, ɪ, ʃ	s(40%), b(20%), ʒ(20%), ʃ(20%)
ø	k	a, j, o	k, o, s	o(100%)
œ	o	o	k, o	o(100%)
ő	an, en, im, n, nt, in	a, e, j, n, s, t, ə, ɨ, ʃ	a, ɨ	ɨ(36%), i(29%), e(21%), a(7%), ə(7%)
ɔ	o, vo, ɔa	o, ɔ	k, o	o(67%), ɔ(22%), ə(11%)
ű	te, ump, un	n, o, t, u	u	u(33%), e(33%), o(17%), ə(17%)
ə	a, e, je	j, s, v, ʃ	a, j, v	e(50%), a(50%)
ɛ	a, aɪe, e, ea, etə, je, jeste, sea, va, ə	a, e, j, s, t, ʃ	a, e, j, o, ə	e(65%), a(17%)
ɛː	ɛː	a	a, e, j	a(100%)
ő	im, in, yʊ, ɔa, ɔam, ɔame, in	a, i, j, n, z	i, ɨ	i(57%), y(14%), a(14%), ə(14%)
g	bə, g	d, e, i, o, r, ŋ, g	k, g	g(50%), t(17%), d(17%), b(17%)
ʁ	r, rə, rə	r	r	r(65%), ɾ(12%)
ʒ	dikə, dʒi, ib, z, gə	d, v, ʒ	d, k, s, t, z, ʃ, ʒ	ʒ(44%), z(11%), j(11%), g(11%), d(11%), b(11%)

Table D.19: Phones predicted by our different models with a confidence above 0.05, from French to Romanian. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

D.5.5 From Romanian

Romanian to	Spanish			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, aɾ	a	a	a(78%), e(9%), ɛ(9%)
b	b, β, βjo, βo	b, p	b, β	β(67%), b(26%)
d	d, da, do, ðo	b, d	d, i, ɛ	d(53%), ð(19%), n(8%), i(6%)
f	f	f, i	a, f, i, o, u	f(90%)
h	b, k	a, b, f, i, k, o, p, t, u, θ	a, e, f, i, j, k, x, g, ɣ, ʝ	k(50%), b(50%)
i	a, e, i, iɾ, ja, je, jɛ, li, ɛ	a, e, i, j, ɛ	e, i, j, l	i(60%), ɛ(20%), e(9%), j(5%)
j	i, j, l, ɛ, ɾ, j, θ	a, e, i, j, o, s, u, ɔ, ɛ, β	a, e, i, j, k, l, ɛ, ɣ, ɪ, ʝ	j(23%), i(18%), ɾ(14%), ʝ(9%), β(9%)
k	ko	k	k	k(74%), ɣ(19%)
l	l	l, ʎ	l	l(80%), ʎ(18%)
m	m, mo	m	m	m(94%)
n	n, njo, no	n	n	n(87%)
o	o, uno, we, ɔ	o, w, ɔ	o, w, ɔ	o(32%), ɔ(25%), e(22%), u(8%), ɛ(8%)
p	p, po	p	p	p(71%), β(16%), t(8%)
r	lo, r, ro, ɔr, ɾ	r, ɾ	r, ɔ, ɾ	ɾ(77%), r(17%)
s	es, s, so	s	s	s(88%)
t	kto, t, te, to, tɾo, ðo	t	t	t(62%), ð(20%), d(5%), θ(5%)
u	o, tu, u, ɔ	o, u, ɔ	o, u, ɔ	u(35%), o(30%), ɔ(25%)
v	b	b	a, b	b(83%), β(8%)
w	u, w, ʎ	o, u, ɔ	o, w, ɔ	w(33%), ɔ(33%), ʎ(33%)
y	a, an, að, en	a	o, u, ɔ	a(75%), e(25%)
z	d, dj, p, s, ðo, ʝ	a, b, d, e, i, o, ɛ, j	d, e, k, n, p, s, t, x, θ	d(33%), s(17%), ɾ(17%), ʝ(17%), ð(17%)
ŋ	n, ŋ, ɛŋ	f, k, m, n, ŋ, g, ɣ, ɲ	m, n, ŋ, ɲ	ŋ(44%), n(22%), ɲ(22%), l(11%)
ɔ	o, u, w, ɔ	o, w, ɔ	o, w, ɔ	w(42%), ɔ(37%), o(16%), u(5%)
ə	a	a, e, i	a, b, e, j, ʝ	a(81%), e(9%)
g	g, go, ɣo	k, g	k, g, ɣ	g(48%), ɣ(29%), ʝ(10%)
i̯	a, e, wa, ɛ	a, e, i, o, ɛ	a, e, i, j, ɛ, ɪ, ʝ	a(55%), ɛ(24%), e(15%)
ɪ	s, xo, ɪ, βiɾ	a, e, i, k, l, o, s, t, u, θ	a, e, i, j, l, x, ɛ, ɣ, ɪ, ʝ	s(25%), o(25%), u(12%), i(12%), ɲ(12%), ɪ(12%)
ɾ	l, r, ɾ	a, e, i, l, o, r, u, ɔ, ɛ, ɾ	l, r, ɾ	ɾ(63%), l(28%)
ʃ	s	k, s, θ	s, θ	θ(56%), s(23%), ʝ(7%)
ʊ̯	o, βo	e, i, o, p, t, u, ɔ, ʊ̯, β	o, u, w, ɔ	o(60%), u(20%), e(13%), ɔ(7%)
ʒ	x, xi, g	a, e, f, i, j, s, u, x, ɛ, θ	k, l, s, x, g, ɣ, ʎ	x(28%), ɾ(28%), t(11%), θ(11%), ʝ(6%), g(6%), ʝ(6%), ð(6%)

Table D.20: Phones predicted by our different models with a confidence above 0.05, from Romanian to Spanish. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

Romanian to	Portuguese			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, aɪ, ɐ	a, ɐ, ɐ̃	a, ɐ	a(58%), ɐ(14%), ɐ̃(8%), ɑ(5%)
b	b, bʊ, v, vʊ	b, v	b, v, g	b(58%), v(35%)
d	d, dɐ, dʊ, ʊ	d	d, f, ʒ	d(68%), ɹ(10%)
f	f	f	f	f(93%)
h	k, v	b, d, k, n, p, t, v, g, ʒ	a, f, i, k, t, v, ɐ, g, ʊ	v(50%), k(50%)
i	e, eɪ, i, noŋ, ɨ	d, i, j, n, o, t, u, ũ, ɪ, ʒ	e, i	i(60%), e(27%)
j	i, l, v, ɛ, ɹ, ʃ	d, e, i, n, o, u, ũ, v, ɨ, ɪ	a, e, i, j, ɐ, ɐ̃, ɛ, g, ʒ	ɹ(25%), i(20%), j(15%), v(10%), ʃ(10%), ʁ(10%)
k	k, kʊ, gʊ	k	k	k(74%), g(21%)
l	l, lʊ	l	l	l(86%)
m	m, meɪ, meɪŋ, mʊ	m	m	m(95%)
n	n, nʊ, ŋ, ʊ	n	n	ŋ(43%), n(33%)
o	o, u, ũŋ, ɔ, ʊ	f, o, u, ũ, ɔ	o, ɔ	o(40%), u(29%), ɔ(20%)
p	bʊ, p, pʊ	p	p	p(82%), b(13%)
r	lʊ, ɹ, rə, rʊ, ʁ, ʁʊ, ʊ	r, ʁ	l, r, ʁ	r(41%), ɹ(39%), ʁ(15%)
s	s, sʊ, zʊ, ʃ	s	s, z, ʃ	s(48%), ʃ(41%), z(8%)
t	dʊ, t, tɨ, tʊ, ɐdʊ	t	t	t(74%), d(25%)
u	bʊ, m, o, om, u, ɔ, ʊ	o, u, ũ, ɔ	o, u, ɔ	u(53%), o(26%), ʊ(5%)
v	v, ʊ	v	b, f, v, ɐ	v(96%)
w	o	o, u, ũ, ɔ	k, o, u, ɔ, ʊ	w(33%), u(33%), n(33%)
y	e, ɐ̃	a, d, ɐ, ɐ̃, g	o, u, ũ, ɔ	e(33%), ɐ̃(33%), ɐ(33%)
z	d, djʊ, dʊ	d, ʒ	d, i, s, z, ɨ, ʃ, ʊ, ʒ	d(67%), z(17%), ɪ(17%)
ŋ	ŋ, nŋ	m, n, ŋ, g, ɪ	e, i, n, ŋ, ɐ̃, ɨ, ʊ	ŋ(100%)
ɔ	o, u, ɔ	o, ɔ	o, ɔ	ɔ(47%), o(33%), u(20%)
ə	ɐ	a, d, v, ɐ, ɐ̃, ɑ, g, ʊ	a, i, ɐ, ɨ, ʊ	ɐ(72%), i(7%)
g	vʊ, g, gʊ	k, g	g	g(88%), v(6%), m(6%)
ɨ	e, i, ɐ, ɐ̃	a, d, e, n, o, u, ũ, ɐ, ɐ̃, ɑ	a, d, e, i, v, ɐ, ɛ, ɨ, ʊ	ɐ̃(47%), ɪ(32%), i(11%), e(5%), ɐ(5%)
ɪ	ʃ, ʎ	a, d, e, i, n, o, u, ɐ̃, ɪ, ʊ	e, i, j, ɐ̃, ɛ	u(17%), ɐ̃(17%), ʃ(17%), r(17%), ɔ(17%), ʎ(17%)
r	b, l, r, ʁ	a, d, e, l, t, u, ɐ, r, ʁ	l, ɛ, r, ʁ	r(64%), ɹ(12%), ʁ(9%), l(6%), b(6%)
ʃ	s, ʃ	s, ʒ	s, ʃ	s(69%), ʃ(21%), z(10%)
ʊ	vʊ, ʊ	d, n, o, t, u, ũ, v, ɐ̃, ɪ, ʊ	k, o, u, ɔ, ʊ	ʊ(50%), u(14%), v(7%), o(7%), ŏ(7%), ɐ̃(7%), ɪ(7%)
ʒ	a, ʒ	s, ʒ	g, ʒ	ʒ(95%), ɹ(5%)

Table D.21: Phones predicted by our different models with a confidence above 0.05, from Romanian to Portuguese. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

Romanian to	French			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, e, o, ə, ɛ	a	a	a(50%), ɛ(18%), e(11%), o(7%), ə(7%)
b	b, v	b, k, ʒ	b, f, k, v	b(40%), v(30%), z(10%), g(10%), ʒ(10%)
d	d	d	d	d(67%), ʁ(17%), g(8%), ʒ(8%)
f	f	f	f	f(100%)
i	i	e, i, j, ɛ, ɛ̃	e, i, j, ə	i(61%), ɛ̃(28%), ɑ̃(11%)
j	a, i, iv, j, l, v, ʁ	e, f, i, j, l, v, ə, ɛ, ʒ	a, e, i, j, w, ɛ	j(29%), l(14%), i(14%), w(7%), v(7%), k(7%), e(7%), ʒ(7%), ʁ(7%)
k	k	k	k	k(87%), ʁ(13%)
l	l	l	l	l(95%)
m	m, mɛ	m	m	m(100%)
n	an, m, n, ɑ̃	n	n	n(69%), t(19%), m(6%), ʁ(6%)
o	j, o, u, w, wa, œ, ɔ	o, u, y, ø, ɔ	ɔ	ɔ(41%), u(12%), o(12%), e(6%), a(6%), ø(6%), œ(6%), ɔ̃(6%), ɛ(6%)
p	p	p	p	p(100%)
r	l, ʁ	ʁ	ʁ	ʁ(88%), l(9%)
s	s, z, ze	s, ʒ	s	s(58%), z(16%), j(11%), ʁ(11%), v(5%)
t	s, t	t	t	t(77%)
u	u, y, ə-	u, y, œ	u, y, ɔ	y(48%), u(28%), ɔ̃(8%)
v	de, v	v, ʒ	f, v	v(85%), f(8%), ʁ(8%)
w	w	u, y	o, y, ɔ	w(100%)
y	y	a, d, k, n, t, ə, g, ʁ, ʃ	o, u, y, ɔ	a(50%), ɛ̃(50%)
z	ʒ	d, ʒ	d, s, ʒ	z(50%), ʒ(50%)
ŋ	ŋ	k, m, n, g	m, n, ɑ̃	
ɔ	ɔ	u, y, ɔ	ɔ	ɔ(100%)
ə	aʁin, wa, ɛ, ʁal	a, e, i, j, l, s, y, ə, ɛ	a, e, i, j, l, ə	a(25%), i(19%), e(12%), ɛ̃(12%), o(6%), ɔ̃(6%), ɑ̃(6%), ɔ(6%), ɛ(6%)
g	g	k, g	g	g(67%), k(17%), ʒ(17%)
ɨ	ɨ	a, e, i, o, t, y, ɔ, ɛ, ʁ	a, e, i, j, ɛ	ɑ̃(86%), i(14%)
ɪ	ɪ	a, i, j, l, u, y, z, ə, ɛ	a, e, i, j, w, ɑ̃, ə, ɛ	j(100%)
r	n, ʁ	a, i, l, o, u, y, ɔ, ɛ, ʁ	ʁ	ʁ(62%), l(25%), n(12%)
ʃ	p, s	s, z, ʃ	s	s(78%), z(11%), j(11%)
ʊ	l	j, k, n, t, u, w, y, œ, ə	o, u, y, ɔ	l(100%)
ʒ	ʒ	ʒ	ʒ	ʒ(67%), z(17%), ʁ(17%)

Table D.22: Phones predicted by our different models with a confidence above 0.05, from Romanian to French. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

Romanian to	Italian			
Phones	SMT	M-NMT+m	M-NMT+m+shared_emb	Gold
a	a, a:re, va:re	a, a:	a, a:	a:(44%), a(37%), e(10%)
b	b, bio, bo, vo	b, b:, f, p, v	b, f, v	b(68%), v(20%)
d	d, do	d	d, f	d(91%)
f	f	f	f	f(98%)
h	v	b, d, f, k, k:, p, s, t:, v, g	a, a:, b, f, k, k:, v, g	v(100%)
i	a, e, i, io, i:re, ro, ɛ	a, a:, e, e:, i, i:, u:, ɛ, ɛ:, ʊ	e, i, i:, j, l, ɛ:	i(50%), i:(19%), e(12%), ɛ(7%)
j	i:re, re, ɛ	a, a:, e, e:, i, j, o, ɔ:, ɛ, ɛ:	a, a:, e, e:, j, o, w, ɔ:, ɛ, ɛ:	r(35%), i:(12%), i(10%), v(8%), j(8%)
k	k, ko	k	k	k(87%), g(9%)
l	l, la, llo	l	l	l(84%), j(8%)
m	m, me, mo	m	m	m(89%)
n	n, ne, no	n	a, m, n	n(86%)
o	o, ɔ, ɔ:	o, o:, u:, w, ɔ, ɔ:	o, o:, ɔ, ɔ:	ɔ:(28%), o(21%), o:(21%), ɔ(19%)
p	p, po	p	m, p	p(85%), t:(5%), p:(5%)
r	lo, r, re, ro, rro	i, j, r	r	r(96%)
s	s, sio, so, zo	s	s	s(76%), z(15%)
t	t, te, to, tro	t, t:	t	t(85%), t:(7%)
u	o, ɔ, ʊ, ʊo	o, o:, u, u:, ɔ, ɔ:, ʊ	o, u, u:, ɔ, ʊ	o(39%), ʊ(17%), u:(10%), ɔ(10%), u(8%), o:(6%), ɔ:(5%)
v	v	b, f, v	a, b, f, v	v(97%)
w	l, ol, u:, ɔ:	o, u, u:, ɔ, ɔ:, ʊ	f, k, o, o:, u, u:, w, ɔ, ɔ:, ʊ	l(100%)
y	a:, e:	a, a:, o	o, o:, u, u:, ɔ, ɔ:, ʊ	e:(50%), a:(50%)
z	d, dj, do, tso	d, f	d, n, s, t, z	d(42%), s(17%), z(8%), z:(8%), j(8%), ʒ(8%), ʒ:(8%)
ŋ	n, ŋ	k, m, n, ŋ, g, ɲ	a, m, n, ŋ	n(64%), ŋ(36%)
ɔ	o, ɔ	o, o:, u:, ɔ, ɔ:, ʊ	o, o:, ɔ, ɔ:	ɔ(52%), ɔ:(22%), o(9%), o:(9%)
ə	a	a, a:, i, o, ɔ:	a, a:, e, l, o, s, v, w	a(72%), e(12%), i(8%)
g	vo, g, go	k, ŋ, g	k, g	g(85%)
i̇	a, i	a, a:, i, o, u, u:, ɔ:	a, a:, e, e:, i, i:, j, ɛ, ɛ:	a(40%), i(29%), u(7%), a:(7%)
ɪ	j, ɪ, ɲjo, ʎo	a, a:, e, e:, i, i:, o, u:, ɛ, ɛ:	a, a:, e, e:, i, i:, j, w, ɛ, ɛ:	o(33%), l(17%), j(17%), ʎ(17%), ɪ(17%)
ɾ	l, r	a, a:, j, o, o:, r, u:, ɔ, ɔ:, ʊ	r	r(71%), l(21%), ɾ(5%)
ʃ	s, tʃo	i, s, t	s, ʃ, ʃ:	ʃ(61%), s(27%)
ʊ	no, o, vo	f, i, i:, m, o, p, u:, v, ɔ, ʊ	f, o, u, w, ɔ, ɔ:, ʊ	o(62%), n(15%), u:(8%), o:(8%), m(8%)
ʒ	dʒ, ʒ	d, s, v	d, k, t, g	ʒ(67%), ʒ:(19%), r(6%)

Table D.23: Phones predicted by our different models with a confidence above 0.05, from Romanian to Italian. In parenthesis, the actual frequency of the different target phones aligned with the input phones in the gold data.

D.5.6 Custom cost function

When aligning phonetized words, we wanted to provide the Needleman-Wunsch algorithm with a custom cost function, reflecting phonetic similarity, and therefore being harsher on an [a] aligned with a [b] than on a [p] aligned with a [b].

Each phone is associated with backness/height values for vowels and place/manner values for consonants. Backness numeric values are 1 for front, 1.5 for near front, 2 for central, 2.5 for near back, and 3 for back. Height numeric values are 0 for close, 0.3 for near close, 0.6 for close mid, 1 for mid, 1.3 for open mid, 1.6 for near open, and 2 for open. Manner numeric values are 1 for nasals, 2 for stops, 3 for affricates, 4 for fricatives, 5 for approximants, etc. Places numeric values are 1 for bilabials, 2 for labiodentals, 3 for labiovelar, 4 for lingualabials, 5 for dentals, 6 for alveolars, 7 for postalveolars, etc.[1]

We consider that if two phones are equal, unaligning them is costly (cost of 15). A 'i' and a 'j' phones are similar too (cost of 7). We convert other phones to their types (Vowel or Consonants), and consider that they are very easy to unalign if they are of different types (cost of 0). For phones (p) of the same type, we apply a scoring S, which, for vowels, is:

$$S(p_1, p_2) = 10 - 5 * (|\text{backness}(p_1) - \text{backness}(p_2)| + |\text{height}(p_1) - \text{height}(p_2)|) \quad \text{(D.1)}$$

[1] Full lists can be found at https://github.com/clefourrier/PLexGen/blob/master/language_definition/phones/pulmonic_consonants.py.

And for consonants:

$$S(p_1, p_2) = 10 - 5 * (|\,\text{place}(p_1) - \text{place}(p_2)| + |\,\text{manner}(p_1) - \text{manner}(p_2)|) \quad \text{(D.2)}$$

D.5.7 Studied phone correspondences

In this subsection, we provide sound correspondence examples that we extracted for our phones of interest. [2]

The file first level of keys is the phonetic context in Latin, with actual letters in small case, uppercase 'C' to indicate 'any consonant', uppercase 'V' 'any vowel', and the dash around the context to indicate if the group is preceded/succeded ('-' before/after) or not (no dash) by other letters. The second level contains 5 keys: 'c' for the character correspondence to the context in Latin's descendants, 'p' for the phonetic correspondence to the context in Latin's descendants, 'example' for a use case, 'origin' for the source (either Dorman (2010) or Boyd-Bowman (1980)), and 'rule' for the generality level of the rule (all consonants for 'general', p, b, v consonants for 'PBV' and t or d consonants for 'TD').

-sC-	
c	IT: '-sC-', ES: '-sC-', PT: '-sC-', FR: '-C-'
p	IT: ['-', 's', 'C', '-'],ES: ['-', 's', 'C', '-'], PT: ['-', 'ʃ', 'C', '-'],FR: ['-', 'C', '-']
example	'pescare, pascar, pescar, pêcher'
origin	'Dorman2010, 9|Bowman1980, 35'
rule	'general'
sC-	
c	IT: 'sC-',ES: 'esC-',PT: 'esC-',FR: 'C-'
p	IT: ['s', 'C', '-'],ES: ['e', 's', 'C', '-'], PT: ['i', 'ʃ', 'C', '-'],FR: ['e', 'C', '-']
example	'spada, espada, espada, épée'
origin	'Dorman2010, 9|Bowman1980, 35'
rule	'general'

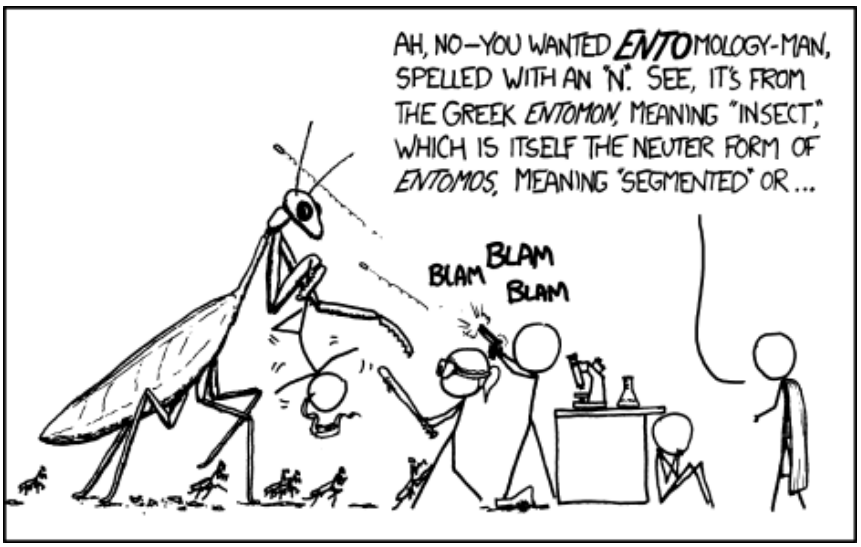
Table D.24: Sample examples of the identified correspondences

[2] The full file is available at `https://github.com/clefourrier/selected-romance-sound-correspondences`.

If you read this far...

So long, and thanks for all the fish!

Adams (1984)



Wrong superhero (XKCD 1012)

RÉSUMÉ

En linguistique historique, les cognats sont des mots qui descendent en ligne directe d'un ancêtre commun, leur proto-forme, et qui sont ainsi représentatifs de l'évolution de leurs langues respectives à travers le temps. Comme ils portent en eux l'histoire phonétique des langues auxquelles ils appartiennent, ils permettent aux linguistes de mieux déterminer toutes sortes de relations linguistiques synchroniques et diachroniques (étymologie, phylogénie, correspondances phonétiques). Les cognats de langues apparentées sont liés par des correspondances phonétiques systématiques. Les réseaux de neurones, particulièrement adaptés à l'apprentissage de motifs latents, semblent donc bien un bon outil pour modéliser ces correspondances. Dans cette thèse, nous cherchons donc à étudier méthodiquement l'applicabilité de réseaux de neurones spécifiques (inspirés de la traduction automatique) à la `prédiction de mots historiques', en nous appuyant sur les similitudes entre ces deux tâches. Nous créons tout d'abord un jeu de données artificiel à partir des règles phonétiques et phonotactiques des langues romanes, que nous utilisons pour étudier l'utilisation de nos réseaux en situation controlée, et identifions ainsi sous quelles conditions les réseaux de neurones sont applicables à notre tâche d'intérêt. Nous étendons ensuite notre travail à des données réelles (après avoir mis à jour une base étymologiques pour obtenir d'avantage de données), étudions si nos conclusions précédentes leur sont applicables, puis s'il est possible d'utiliser des techniques d'augmentation des données pour pallier aux manque de ressources de certaines situations. Enfin, nous analysons plus en détail nos meilleurs modèles, les réseaux neuronaux multilingues. Nous confirmons à partir de leurs résultats bruts qu'ils semblent capturer des informations de parenté linguistique et de similarité phonétique, ce qui confirme des travaux antérieurs. Nous découvrons ensuite en les sondant (probing) que les informations qu'ils stockent sont en fait plus complexes : nos modèles multilingues encodent en fait un modèle phonétique de la langue, et apprennent suffisamment d'informations diachroniques latentes pour permettre à des décodeurs de reconstruire la proto-forme (non vue) des langues étudiées aussi bien, voire mieux, que des modèles bilingues entraînés spécifiquement sur cette tâche. Ces informations latentes expliquent probablement le succès des méthodes multilingues dans les travaux précédents.

MOTS CLÉS

Traitement Automatique des Langues (TAL), linguistique computationnelle, étymologie computationelle, reconstruction de mots historiques, cognats, traduction automatique neuronale

ABSTRACT

In historical linguistics, cognates are words that descend in direct line from a common ancestor, called their proto-form, and therefore are representative of their respective languages evolutions through time, as well as of the relations between these languages synchronically. As they reflect the phonetic history of the languages they belong to, they allow linguists to better determine all manners of synchronic and diachronic linguistic relations (etymology, phylogeny, sound correspondences). Cognates of related languages tend to be linked through systematic phonetic correspondence patterns, which neural networks could well learn to model, being especially good at learning latent patterns. In this dissertation, we seek to methodically study the applicability of machine translation inspired neural networks to historical word prediction, relying on the surface similarity of both tasks. We first create an artificial dataset inspired by the phonetic and phonotactic rules of Romance languages, which allow us to vary task complexity and data size in a controlled environment, therefore identifying if and under which conditions neural networks were applicable. We then extend our work to real datasets (after having updated an etymological database to gather a correct amount of data), study the transferability of our conclusions to real data, then the applicability of a number of data augmentation techniques to the task, to try to mitigate low-resource situations. We finally investigat in more detail our best models, multilingual neural networks. We first confirm that, on the surface, they seem to capture language relatedness information and phonetic similarity, confirming prior work. We then discover, by probing them, that the information they store is actually more complex: our multilingual models actually encode a phonetic language model, and learn enough latent historical information to allow decoders to reconstruct the (unseen) proto-form of the studied languages as well or better than bilingual models trained specifically on the task. This latent information is likely the explanation for the success of multilingual methods in the previous works.

KEYWORDS

Natural Language Processing (NLP), computational linguistics, computational etymology, historical words reconstruction, cognates, Neural Machine Translation (NMT)