A Neural Approach for Detecting Morphological Analogies

Safa Alsaidi, Amandine Decker, Puthineath Lay, Esteban Marquer, Pierre-Alexandre Murena, Miguel Couceiro

A Neural Approach for Detecting Morphological Analogies

Safa Alsaidi*
IDMC, *Université de Lorraine*
Nancy, France
safaa-98@hotmail.com

Amandine Decker*
IDMC, *Université de Lorraine*
Nancy, France
decker.amandine@hotmail.com

Puthineath Lay*
IDMC, *Université de Lorraine*
Nancy, France
puthineathlay@gmail.com

Esteban Marquer
Université de Lorraine, CNRS, LORIA
F-54000, France
esteban.marquer@loria.fr

Pierre-Alexandre Murena
HIIT, Aalto University
Helsinki, Finland
pierre-alexandre.murena@aalto.fi

Miguel Couceiro
Université de Lorraine, CNRS, LORIA
F-54000, France
miguel.couceiro@loria.fr

***Abstract*—Analogical proportions are statements of the form "A is to B as C is to D" that are used for several reasoning and classification tasks in artificial intelligence and natural language processing (NLP). For instance, there are analogy based approaches to semantics as well as to morphology. In fact, symbolic approaches were developed to solve or to detect analogies between character strings, e.g., the axiomatic approach as well as that based on Kolmogorov complexity. In this paper, we propose a deep learning approach to detect morphological analogies, for instance, with reinflexion or conjugation. We present empirical results that show that our framework is competitive with the above-mentioned state of the art symbolic approaches. We also explore empirically its transferability capacity across languages, which highlights interesting similarities between them.**

***Index Terms*—morphological analogy, semantic analogy, deep learning, analogy classification**

I. Introduction

An analogy, or analogical proportion, is a relation between four elements A, B, C, and D meaning "A is to B as C is to D", often written as $A : B :: C : D$. A typical example would be the semantic analogy "*kitten* is to *cat* as *puppy* is to *dog*".

Analogy is a major component of human cognition [1, 2] and of the process of learning by extrapolation [3], and multiple cognitive linguistics theories support the importance of analogies in learning and using a language [4, 5, 6]. Analogies have been extensively studied in NLP, resulting in different formalizations (like the one described in Section II-A) with noteworthy applications such as derivational morphology [7, 8] and machine translation [9].

Analogical reasoning can capture different aspects of the manipulated objects. For example, analogies on words can refer exclusively to their morphology ("*cats* is to *cat* as *trees* is to *tree*") or their semantics ("*kitten* is to *puppy* as *cat* is to *dog*"). The question of the correctness of an analogy $A : B :: C : D$ is a difficult task and it has been tackled using both formal [7, 10] and empirical approaches [8, 11, 12, 13, 14]. Although the original challenge is to find some structure from A, B, C, and D, recent empirical works propose data-oriented strategies based on machine learning to learn the correctness of analogies from past observations. In particular, Lim *et al.* [11] propose a deep learning approach to train models of analogies on corpora of semantic analogies and pretrained GloVe embeddings [15]. This approach achieves competitive results on analogy classification and completion.

In this paper, we adapt the approach developed by Lim *et al.* [11] for semantic word analogies to morphological analogies. The major difference between the two approaches is that ours relies on a morphological word embedding, which had to be entirely developed and trained. Indeed, the pretrained embeddings used in [11] are adapted to semantic analogies only, and there are currently no readily available morphological word embeddings. Furthermore, we explore the potential of transferring our neural analogy model across languages.

Our method achieves competitive results to state of the art symbolic approaches, and obtains promising results in transferability settings. This shows the potential of Lim *et al.*'s approach [11] for applications other than semantic analogies. We illustrate this fact with morphological analogies, and that opens the possibility of tackling multimodal analogies. As long as we have a fitting embedding model, this approach could be further extended to analogies between domains that are usually considered too complex, such as analogies between functions or other complex models.

This paper is organized as follows. We recall in Section II a simple formalization of analogy and present related works, in particular, the approach of Lim *et al.* [11] by which our framework is inspired. In Section III we explain how we adapt this approach to morphological analogy. An empirical study carried out on the Sigmorphon2016 [16] and Japanese Bigger Analogy [17] datasets is presented in Section IV, in which we compare our results to other state of the art approaches to morphological analogy, that show the competitiveness of our approach. We also explore how the models trained on each language perform when transferred to the other languages in Section V, that reveal interesting similarities between them.

*Authors contributed equally.

II. Related Works

In this section we briefly discuss the axiomatic setting of analogy and address two main problems dealing with analogical reasoning, namely analogical identification (or classification) and inference (or regression) (Section II-A). We then discuss different approaches for modeling analogies on words, from a morphological (Section II-B) and semantic (Section II-C) point of view.

A. Formal Analogy

Multiple attempts have been made to formalize and manipulate analogies, starting as early as De Saussure's work in 1916 [4]. We refer to [18] for an introduction to analogy in linguistics. Most of the recent works on analogy use a formalization based on that of [19], which is related to the common view of analogy as a geometric (Equation (1)) or arithmetic proportion (Equation (2)) or, in geometrical terms, as a parallelogram in a vector space (Equation (3)).

$$\frac{A}{B} = \frac{C}{D} \tag{1}$$

$$A - B = C - D \tag{2}$$

$$\vec{A} - \vec{B} = \vec{C} - \vec{D} \tag{3}$$

This formalism takes an axiomatic approach to define *proportional analogies* [10, 20]: four objects A, B, C, and D are in *analogical proportion* (*i.e.* $A : B :: C : D$) if and only if the following three postulates hold true:

- $A : B :: A : B$ (reflexivity);
- $A : B :: C : D \rightarrow C : D :: A : B$ (symmetry);
- $A : B :: C : D \rightarrow A : C :: B : D$ (central permutation).

These postulates imply several other properties:

- $A : A :: B : B$ (identity);
- $A : B :: C : D \rightarrow B : A :: D : C$ (inside pair reversing);
- $A : B :: C : D \rightarrow D : B :: C : A$ (extreme permutation).

It can be easily verified that the geometric and arithmetic proportions (Equations (1) and (2)) and the parallelogram rule (Equation (3)) define proper analogies. Another frequently accepted postulate is uniqueness: for a triple $\langle A, B, C\rangle$, if there exists D such that $A : B :: C : D$ then D is unique, which entails that $A : A :: B : C \rightarrow C = B$. Most of the works on analogy relying on such a formalization concern analogy between symbols and strings of symbols ($abc : abd :: efg : efh$), which can be directly related to morphology ($cat : cats :: dog : dogs$). An example out of this scope is [21] which deals with boolean functions.

Analogy related tasks can be divided in two main axes:

- *analogy identification*, in which given a quadruple $\langle A, B, C, D\rangle$ we want to know whether $A : B :: C : D$ is a valid analogy [7, 11]; this can be extrapolated to analogy extraction from sets of objects as in [7];
- *analogical inference*, which aims at solving "analogical equations", *i.e.*, finding the indeterminate x in expressions of the form $A : B :: C : x$ [8, 11, 22].

When the manipulated objects have a vector representation (typically with embeddings), analogy identification can be seen as a binary classification task, whereas analogy solving can be seen as a regression task. In this paper, we focus on analogical classification that can have a wide variety of linguistic applications as illustrated in [7].

B. Approaches to Model Analogies Between Symbols and Morphological Analogies

Most algorithmic approaches to solving morphological analogies rely on the aforementioned formal characterization of proportional analogies. Recent approaches include those by Fam and Lepage (2018) [7] and the `Alea` algorithm proposed by Langlais *et al.* (2009) [22]. Fam and Lepage's approach relies on feature vectors to detect analogies within a list of words and create analogical grids, allowing to build analogies between more than 4 words. The axioms of Lepage [18] have been also reformulated by Yvon [23] to give a closed form solution, which is computed by `Alea` in a Monte-Carlo setting. In practice, `Alea` uses random slicing and merging of the character strings A, B and C to obtain potential solutions to $A : B :: C : x$, which are then ranked by how frequently they appear when repeating the random process a certain number of times. Good results can be obtained with 1000 receptions [22].

A more empirical approach was proposed by Murena *et al.* (2020) [8], which relaxes the formal definition of analogical proportion. Following preliminary evidences that humans may follow a simplicity principle when solving analogies [24, 25, 26], the authors propose to solve analogical equations $A : B :: C : x$ by finding the x that minimizes the total description length (or Kolmogorov complexity) of $A : B :: C : x$. The total description length is evaluated using a simple description language for character strings and an associated binary code. Both `Alea` and Fam and Lepage's approaches were shown to be outperformed by Murena *et al.*'s Kolmogorov complexity approach on analogies, on the Sigmorphon2016 dataset.

C. Approaches to Model Semantic Analogies

There is a long tradition of works on word representation using statistical methods based on co-occurrence of words to represent them as vectors, that we usually call *word embeddings*. Since early works on embedding spaces [13], it is usually assumed that embeddings represent the semantics of words and that analogies in the vector space can be defined by simple arithmetic operations, either by relying on the parallelogram rule [14] (see Equation (4)) or by using a cosine distance [14] such as Equation (5) or *3CosMul* [12]; other similar metrics can be found in [27].

$$\vec{B} - \vec{A} + \vec{C} = \vec{D} \tag{4}$$

$$\frac{(\vec{A} - \vec{B}) \cdot (\vec{C} - \vec{D})}{||\vec{A} - \vec{B}|| \times ||\vec{C} - \vec{D}||} \tag{5}$$

For instance, GloVe or *word2vec* [13] embeddings are usually able to solve simple analogical equations using the parallelogram rule such as $\vec{cat} - \vec{kitten} + \vec{dog} \approx \vec{puppy}$.

However, as mentioned in [14], those simplistic rules for analogy do not match human performance in all cases. Additionally, some strict properties of formal analogy can contradict human intuition, like *uniqueness* which is problematic when dealing with (quasi-)synonyms, which should be interchangeable solutions of analogical equations on word semantics. Hence, it becomes necessary to relax the model of analogy to match human performance on analogies. While Murena *et al.* [8] follow a simplicity principle on the transformation between words, Lim *et al.* [11] propose to learn an analogy operator directly from analogies as an elegant way to get out of the simplistic models of parallelogram and cosine analogy. They use simple neural networks to learn analogies from a dataset of semantic analogies, and propose different models for classification and regression purposes. Moreover, they rely on the properties of formal analogies mentioned in Section II-A to increase the amount of training data (as described in Section III-C) and to ensure that models fit those properties to a certain degree.

III. Approach

We adapt the approach developed by Lim *et al.* [11] for semantic analogies to apply it to morphological analogies. Even though we use the same architecture for analogy classification (see Section III-A) and a similar training process (Section III-C), our approach differs in the use of a custom embedding model (Section III-B, based on [28]) trained along the classifier.

A. Simple Neural Model for Classification

The architecture proposed in [11] for classifying analogies is a Convolutional Neural Network (CNN) taking as input the embeddings of size n of four elements A, B, C, and D stacked into a $n \times 4$ matrix. The CNN has 3 layers as follows (see Figure 1).

- A first convolutional layer with filter of 1×2 is applied on the embeddings, such that for each component $\cdot_i$ of the embedding vector, each filter spans across A_i and B_i simultaneously, and across C_i and D_i simultaneously, with no overlap. In this architecture, 128 such filters are used, with a Regularized Linear Unit (ReLU) as activation function. This layer can be seen as extracting the relations $A : B$ and $C : D$.
- A second convolutional layer with 64 filters of 2×2 is applied on the resulting $128 \times n \times 2$ matrix, after which the result is flattened into a $64(n-1)$ unidimensional vector. This layer can be seen as comparing $A : B$ and $C : D$, and acts as the relation "as" in "A is to B as C is to D". Similarly to the previous layer, the activation function is ReLU.
- In the third layer, the output of the second one is fed to a fully-connected layer (or feed-forward layer or perceptron) with a single output that we bind between 1 and 0 using a sigmoid activation.

B. Embedding Model

The model from [11] relies on an extensively pretrained model to produce a vector representation of each word, called its *word embedding*. The analogy model takes the embeddings as inputs, and is thus dependent on the quality and information captured by the embeddings.

In their article, Lim *et al.* [11] work with semantic analogies on English words, and use a pretrained GloVe embeddings that are supposed to encode the semantics of words with high fidelity. However, we work on morphological analogies and on languages which do not necessarily have large embedding models readily available (see Section IV-A), as most embedding models are trained on English. In addition to the difficulty of obtaining embedding models for the languages we work on, we expected semantic embeddings to perform very poorly in a purely morphological task, even if for languages like Chinese where morphology is strongly related to word semantic, that effect might be less important.

For example for German, one of the most readily available languages of the Sigmorphon2016 dataset, we initially experimented with a pretrained German GloVe model[1]. However, this rather large GloVe model only has a coverage of 51.26% (8872 of 17308 words) of the words present in the training set and 49.10% (9798 of 19955 words) for the test set. This means a large portion of the analogies have at least one word unknown by the model, which is thus replaced by an embedding full of 0. Similar issues happened with other pretrained embeddings we tried. Our other preliminary experiments using GloVe and Bert Multilingual [29] confirmed the poor performance of such embedding methods for extensive morphological analogy even if some marginal results can be observed. Due to these issues we will not compare to them further in this article.

To compensate for this issue, we use a character-level word embedding model able to capture the morphological aspect of words. To our knowledge no such embedding model is available for the languages we manipulate, so we decided to train an embedding model together with our classifier. We use a very simple model architecture based on Convolutional Neural Network (CNN) as described in [28]. This architecture is designed to embed individual words for morphological tasks. It is composed of two layers as follows, also presented in Figure 1.

- First we have a character embedding layer, which provides a learnable vector representation of size m for each character. If a character is not seen during training, it's embedding is full of 0 by default. Additionally, we add a special "start of word" and an "end of word" character at each end of the word. For a word of $|w|$ characters, we obtain a $(|w|+2) \times m$ matrix.
- Then, we have a convolutional layers with 16 filters of each of the sizes in $2 \times m, 3 \times m, \cdots, 6 \times m$, for a total of $16 \times 5 = 80$ filters. The filters are used on the

[1] https://int-emb-glove-de-wiki.s3.eu-central-1.amazonaws.com/vectors.txt, from https://deepset.ai/german-word-embeddings

embeddings such that they overlap along the character dimension. Whenever necessary, 0 padding is used such that the center of each filter reach the extremities of the word, as shown in Figure 1. We expect those filters to capture morphemes[2] of up to 6 characters.
- In the output layer, we use a max-pooling method to reduce the output of the convolutional layer along the character dimension, which gives us a vector of 80 components (one component per filter) used as the embedding.

C. Training Process

The model was trained for binary classification using *binary cross entropy*, where the positive class is analogies and the negative is non-analogies. Since the Sigmorphon2016 dataset we use in our experiments contains only positive examples (*i.e.* correct analogical quadruples), we use the same data augmentation process as in [11] to generate negative examples (*i.e.* invalid analogical quadruples).

This process was designed based on the properties of formal analogies, resulting in 8 equivalent forms for every analogy $A : B :: C : D$:

- $A : B :: C : D$ (base form);
- $C : D :: A : B$ (symmetry);
- $A : C :: B : D$ (central permutation);
- $B : A :: D : C$ (inside pair reversing);
- $D : B :: C : A$ (extreme permutation);
- $D : C :: B : A$ (inside pair reversing followed by symmetry);
- $C : A :: D : B$ (central permutation followed by inside pair reversing);
- $B : D :: A : C$ (extreme permutation followed by inside pair reversing).

In addition to those equivalent forms, we have 3 analogical forms per valid analogy which are considered invalid analogies as they cannot be deduced from the base form $A : B :: C : D$ without contradicting at least one of the postulates mentioned in Section II-A:

- $B : A :: C : D$;
- $C : B :: A : D$;
- $A : A :: C : D$.

Using this process we obtain a total of 8 positive and $3 \times 8 = 24$ negative examples per analogy in the dataset, as each negative form has 8 equivalent forms.

To train the embedding and classification model together for each analogy $A : B :: C : D$, we first compute the embedding of A, B, C, and D and train the classifier for all the permutations mentioned above. Additionally, during training we do not apply the negative forms on each permutation of the analogy but only on the base form, resulting in 8 positive examples for 3 negative ones (instead of 8 and 24). This ensures stability of the training process as, for example, it prevents the embedding model from specializing on negative examples as those would be 3 times more frequent in the training data.

[2]Morphemes are minimal morphological units which in our setup can be seen as n-grams.

IV. Experiments

In this section we detail our experimental setup on the Japanese Bigger Analogy Test Set [17] and the Sigmorphon2016 dataset [16] (both detailed in Section IV-A). We also present how we used the approaches from [8, 22] as baselines (Section IV-B).

The code used for our experiments is written in Python 3.9 and PyTorch and is available in the repository https://github.com/AmandineDecker/nn-morpho-analogy. Experiments presented in this paper were carried out using computational clusters equipped with GPU from the Grid'5000 testbed (see https://www.grid5000.fr). Namely, the *grele* and *grue* clusters of were used.

A. Dataset

For our experiments, we use two datasets: the Sigmorphon2016 [16] dataset and the Japanese Bigger Analogy Test Set [17][3].

1) Sigmorphon2016: The Sigmorphon2016 dataset [16] is a large inflexion dataset over 10 languages: Arabic (Romanized), Finnish, Georgian, German, Hungarian, Maltese, Navajo, Russian, Spanish, and Turkish. Most of those languages are considered to have rich inflexions. It is separated in 3 subtasks: inflexion, reinflexion and unlabled reinflexion. In our experiments, we focus on the data from the inflexion task, which is composed of triples of a source lemma A (ex: "do"), a set of features F (ex: present participle) and the corresponding inflected form B (ex: "doing"). For example from the Finish data we have the triple: $A =$"lenkkitossut", $F =$"pos=N,case=ON+ESS,num=PL" (the transformation corresponds to the nominative to essive cases of a noun, for the plural) and $B =$"lenkkitossuilla".

To obtain morphological analogies from this dataset, for any pair of triples $\langle A, F, B\rangle, \langle A', F', B'\rangle$ that have the same set of features ($F = F'$), we consider $A : B :: A' : B'$ an analogical proportion. Note that for each pair of triple, we only generate one analogy, *i.e.*, if we generate $A : B :: A' : B'$ we do not generate $A' : B' :: A : B$ as it will be generated by the data augmentation process. Also, analogies of the form $A : B :: A : B$ will be generated as the set of features is the same ($F = F$). The number of analogies obtained by this process is detailed in Table I. During training and evaluation we generate for each analogy all the permutations and negative examples as mentioned in Section III-C, resulting in a total of 8 valid analogies and repressively 3 and 24 invalid analogies for training and evaluation.

Several approaches were proposed to solve these subtasks, they can be grouped into three trends as mentioned in [16]: NLP pipelines using edit operations, neural approaches, and approaches based on linguistic heuristics. None of the systems originally submitted relied on analogy solving, which is the

[3]https://vecto.space/data/

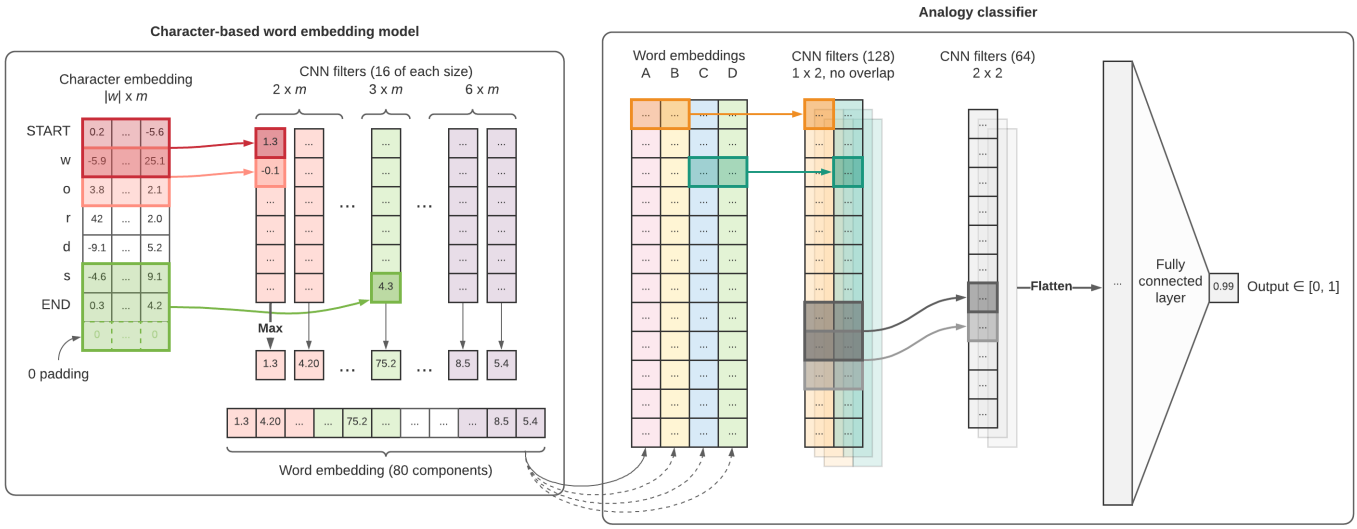


Fig. 1: Overall structure of the CNN character-level word embedding model and CNN classifier.

approach we explore in this paper, in a similar manner as the approach to the task developed in [8].

TABLE I: Number of analogies for each language before data augmentation. We do not use the development set in our experiments.

Language	Train	Dev	Test
Arabic	373240	7671	555312
Finnish	1342639	22837	4691453
Georgian	3553763	67457	8368323
German	994740	17222	1480256
Hungarian	3280891	70565	66195
Maltese	104883	3775	3707
Navajo	502637	33976	4843
Russian	1965533	32214	6421514
Spanish	1425838	25590	4794504
Turkish	606873	11518	11360

2) Japanese Bigger Analogy Test Set: The Japanese Bigger Analogy Test Set contains pairs of words related by a particular morphological relation, each relation is separated in a particular file. From the available relations we focus on inflectional and derivational morphology, and process the data in a similar manner as Sigmorphon2016: for every two pairs of words $\langle A, B\rangle, \langle A', B'\rangle$ related by the same relation we generated an analogical proportion $A : B :: A' : B'$, and we only generate one analogy per pair of word pairs. The selected relations correspond to 1017 pairs of words and 26410 analogies before augmentation.

3) Data for Training and Testing: The model trained on Japanese was trained on 18487 randomly selected analogies (70%) and tested on the remaining 7923. On Sigmorphon2016, as the amount of available data is very large, the models of each language were trained on 50000 and tested on 50000 analogies randomly selected from the train and test data respectively. After data augmentation, this correspond to 400000 positive and 150000 negative examples for training,

TABLE II: List of relations between the words of the Japanese dataset and corresponding number of unique analogies before data augmentation.

	Relation	Example	Pairs
Inflectional morphology	verb_dict - mizenkei01	会う → 会わ/あわ	50
	verb_dict - mizenkei02	出る → 出よ/でよ	51
	verb_dict - kateikei	会う → 会え/あえ	57
	verb_dict - teta	会う → 会っ/あっ	50
	verb_mizenkei01 - mizenkei02	会わ → 会お/あお	50
	verb_mizenkei02 - kateikei	会お → 会え/あえ	57
	verb_kateikei - teta	会え → 会っ/あっ	50
	adj_dict - renyokei	良い → 良く/よく	50
	adj_dict - teta	良い → 良かっ/よかっ	50
	adj_renyokei - teta	良く → 良かっ/よかっ	50
Derivational morphology	noun_na_adj + ka	強 → 強化/きょうか	50
	adj + sa	良い → 良さ/よさ	50
	noun + sha	筆 → 筆者/ひっしゃ	50
	noun + kai	茶 → 茶会/ちゃかい	50
	noun_na_adj + kan	同 → 同感/どうかん	50
	noun_na_adj + sei	毒 → 毒性/どくせい	52
	noun_na_adj + ryoku	馬 → 馬力/ばりき	50
	fu + noun_reg	利 → 不利/ふり	50
	dai + noun_na_adj	事 → 大事/だいじ	50
	jidoshi - tadoshi	出る → 出す/だす	50
	Total		1017

and 1200000 negative examples for testing. However, Maltese, Navajo and Turkish were tested on only 3707, 4843 and 11360 analogies respectively due to the smaller size of the test dataset for those languages.

To maintain a reasonable training time, we trained the models for 20 epochs which correspond to around 6h with our computational resources.

4) Lepage's analogies from Sigmorphon2016: For some of our comparability experiments we use the analogies extracted from Sigmorphon2016 [16] by Lepage [30], as it is the dataset used to compare the baselines in [8]. This dataset contains

analogies extracted from Sigmorphon2016 Task 1 training data by aligning features in a similar manner as what we do. However, for each analogy $A : B :: C : D$, the dataset also contains the following permutations:

- $D : B :: C : A$;
- $D : C :: B : A$;
- $C : A :: D : B$.

We notice that not all the permutations we use (detailed in Section III-C) are considered. Table III presents the portion of the analogies appearing in our dataset that is also appearing in Lepage's dataset (that we call coverage by Lepage's dataset of ours), as well as the converse coverage by our dataset of Lepage's. For our computations, we considered that if any permutation (among the 8 mentioned in Section III-C) of an analogy is present in the other dataset, then the dataset covers this analogy. Surprisingly, neither version completely covers the other, which means that even though our version tends to cover most of Lepage's version, the latter somehow contains some analogies that were not extracted using our process. However, as not much information is provided in [30] about the extraction process, we were not able to determine how our processes differ in, other than the data augmentation process which was circumvented in our coverage computations.

TABLE III: Coverage (in %) between Lepage's version of Sigmorphon2016 and the training set of our version. The coverage of the test set is 0% in both directions for all languages and was not included.

Language	Coverage by Lepage's version of ours	Coverage by our version of Lepage's
Arabic	26.73	60.26
Finnish	21.51	92.30
Georgian	68.22	78.51
German	68.07	92.38
Hungarian	33.14	37.39
Maltese	16.70	61.82
Navajo	7.09	10.18
Russian	25.81	91.81
Spanish	54.23	91.44
Turkish	29.12	71.80

B. Baselines

To compare the performance of our model, we use the methods presented in [8], [7], and [22]. However, as the models from [8, 22] are not designed for classification of quadruples into analogies and non-analogies, we perform several adaptations.

1) Kolmogorov Complexity Based Approach: We use the generation model proposed by Murena *et al.* [8] as a first baseline. As the minimal-complexity approach is designed as a generative model, to have a classification for a given analogy $A : B :: C : D$ we generate the top k most likely solutions for $A : B :: C : x$, in increasing order of Kolmogorov complexity. If D is within the solutions, we consider $A : B :: C : D$ as a valid analogy, otherwise we consider it a non-analogy. We use multiple threshold values from 1 (only the most likely answer) to 10, resulting in multiple baselines. As the accuracy for $1 < k < 10$ is strictly within the range of the accuracy for $k = 1$ and $k = 10$, only $k = 1$ and $k = 10$ are displayed in the results, that we refer to as `Kolmo@1` and `Kolmo@10`.

2) Alea: We use the `Alea` algorithm [22] as a second baseline. This approach relies on a random process to generate potential permutations. For each quadruple $A : B :: C : D$, we generate $\rho = 1000$ potential solutions for $A : B :: C : x$ and select the k most frequent ones. This approach is a generation model like the approach based on Kolmogorov complexity, so we use a similar process of generating potential solution and searching the expected D among them. Similarly to with the Kolmogorov complexity based approach, we use multiple threshold values from 1 (only the most likely answer) to 10, and report only the results with $k = 1$ and $k = 10$, that we refer to as `Alea@1` and `Alea@10`.

3) Lepage Classifier: We use the analogy classifier (`is_analogy` in `nlg/Analogy/tests/nlg_benchmark.py`) from Fam and Lepage's toolset [7] to classify analogies in the same manner as with our deep learning model. This baseline uses a matrix based approach to align characters between words and find common sub-words, and then determine whether there is an analogy. We refer to this baseline as `Lepage`.

C. Results

The classification accuracy for the baselines and our neural classification model (that we refer to as `CNN`) are presented in Table IV.

The Kolmogorov complexity and Alea baselines are slower on some languages like Spanish and German than on other languages. In fact, both approaches are extremely slow on Finnish: while for other languages computations finish (for 50000 examples) in less than 24h, the Kolmogorov complexity approach processes about 4000 examples of the Finnish data in that timeframe whereas Alea is even slower (only 300 analogies processed). For those two baselines we did not compute with a threshold of $k = 10$ on Finnish as the computation time was too large. Further analysis of the specificity of the words involved in the analogies (see Table VI) confirms that the total length of the manipulated words and the number of repeated adjacent letters are correlated with this reduced speed of the Kolmogorov complexity and Alea baselines. In particular, Finnish and Navajo have a high number of adjacent repeated letters, and both Finnish and German have considerably large words. However, while both German and Navajo are slower than other languages, only Finnish is extremely slow. Thus, it is very likely that both characteristics must be present in the words manipulated in order to entail long computation time.

For our three baselines, we notice a strong imbalance of performance between positive and negative examples. Analysis of where the model fails also leads us to testing the baselines on the analogies before applying the augmentation process (see Section III-C), with the results reported in the "Base" column of Table IV for the Kolmogorov complexity approach. For the other approach the results were not significantly different (here and after, by significant we mean that the p-value of student's t test is lower than 0.05) from those of the positive

TABLE IV: Accuracy results (in %) for the classification task for positive (+) and negative (−) examples, as well as examples before augmentation (Base) for the Kolmogorov complexity based approach. Italicized values were obtained over a smaller subset of the data due to the relative slowness of the approaches.

	CNN		**Kolmo@1**			**Kolmo@10**			**Alea@1**		**Alea@10**		**Lepage**	
Language	+	−	Base	+	−	Base	+	−	+	−	+	−	+	−
Arabic	**99.89**	97.52	42.59	28.94	**97.79**	45.73	33.55	97.68	28.28	97.68	34.21	97.68	32.94	97.67
Finnish	**99.44**	82.62	*28.45*	*22.82*	*98.12*	–	–	–	*21.83*	***98.78***	–	–	25.60	98.05
Georgian	**99.83**	91.71	91.68	80.19	95.01	92.41	93.20	94.61	86.81	**95.21**	91.68	95.21	89.78	95.18
German	**99.48**	89.01	85.79	55.61	96.84	86.35	60.27	96.65	85.46	**97.19**	86.90	97.18	83.32	96.86
Hungarian	**99.99**	**98.81**	36.19	31.21	98.40	37.79	36.80	98.23	35.79	98.24	36.58	98.24	33.81	98.25
Maltese	**99.96**	**77.83**	77.99	68.84	69.29	82.03	73.64	67.32	74.49	67.35	78.05	67.32	73.63	67.32
Navajo	**99.53**	90.82	18.54	17.97	**94.93**	24.34	21.45	94.45	16.16	94.76	18.34	94.76	17.35	94.76
Russian	**97.95**	79.85	44.28	33.37	93.66	45.70	36.43	93.30	42.00	93.74	42.37	93.72	41.02	**93.88**
Spanish	**99.94**	78.33	83.62	73.86	86.59	84.24	81.54	86.44	85.23	86.13	85.90	86.13	83.25	**86.62**
Turkish	**99.48**	**92.63**	43.63	39.37	91.40	46.83	43.51	90.78	42.53	90.80	44.76	90.80	34.92	90.80
Japanese	**99.99**	**98.65**	3.82	18.62	98.13	3.85	19.20	98.09	3.75	98.14	3.77	98.14	3.74	98.16

TABLE V: Accuracy (in %) of our model on Lepage's analogies from Sigmorphon2016. The baseline results are taken from [8], as regression accuracy corresponds to classification accuracy when using a regression model to classify analogies.

Language	**CNN**	**Best baseline according to [8]**
Arabic	**98.75**	93.33 (Lepage)
Finnish	93.57	**93.69** (Kolmo)
Georgian	**99.56**	99.35 (Kolmo)
German	**99.56**	98.84 (Kolmo)
Hungarian	**99.32**	95.71 (Kolmo)
Maltese	**97.93**	96.38 (Kolmo)
Navajo	**99.82**	86.87 (Lepage)
Russian	**99.61**	97.26 (Lepage)
Spanish	**97.37**	96.73 (Kolmo)
Turkish	**99.77**	89.45 (Kolmo)

TABLE VI: Particularities (mean ± std. (max)) of the words involved in the analogies.

Language	**Average number of adjacent repeated letters**	**Average word length**
Arabic	0.280 ± 0.463 (2)	8.22 ± 2.23 (20)
Finnish	1.036 ± 0.908 (6)	11.01 ± 3.63 (44)
Georgian	0.022 ± 0.148 (2)	8.11 ± 2.21 (21)
German	0.204 ± 0.432 (3)	10.37 ± 3.47 (35)
Hungarian	0.159 ± 0.384 (2)	8.69 ± 3.03 (20)
Maltese	0.663 ± 0.646 (3)	9.43 ± 3.55 (19)
Navajo	0.851 ± 0.695 (2)	7.69 ± 2.10 (17)
Russian	0.080 ± 0.278 (2)	8.81 ± 2.83 (22)
Spanish	0.090 ± 0.293 (2)	8.94 ± 2.31 (33)
Turkish	0.044 ± 0.206 (1)	8.80 ± 3.61 (25)
Japanese	0.029 ± 0.174 (2)	4.99 ± 3.19 (18)

examples, so we did not report them in Table IV. The results on those "raw" analogies are significantly higher than those on the positive examples for the Kolmogorov complexity baseline, which hints that this approach is not resilient to the permutations performed when augmenting data. In particular, it tends to fail when central permutation is involved. For further comparability with the three baselines, we report in Table V the results of our approach when tested on the analogies extracted by Lepage [30] (see Section IV-A4). In Table V we also showcase the best accuracy results reported in [8] together with the corresponding model. Although results were computed as "the proportion of correct answers when solving analogies" [8], this corresponds to accuracy results when using regression models as classifiers. On all languages our model outperforms the baselines, except for Finnish where the performance is comparable.

Moreover, we could expect the performance for the raw positive examples ("Base" column) to match the one reported in [8], even if both versions of the dataset do not overlap exactly (see Section IV-A). A closer look at the suggested answers shows that this difference is due to a different preprocessing applied to the Sigmorphon2016: while we use all the analogies we can build using our simple alignment of features, Murena *et al.* [8] use a variant where all the analogies that cannot be inferred by relying on only the source terms have been removed (for example in Finnish, the correct answer to the analogy "mäyrä:mäyrässä::kolo:kolossa" is dictated by the rules of vowel harmony). This means that some of the analogies in our version of the dataset are impossible to solve without additional data nor knowledge of the language (neither by an algorithm nor by a human). While the symbolic approaches of [7, 8, 22] do not rely on the languages they manipulate, our approach is trained specifically for a language, which allows it to infer the underlying rules and to perform well even for those analogies. The proportion of such analogies varies from one language to another (according to the selection of the analogy, but also to the rules of the language itself), but no precise measurement was done as it would most likely require a manual analysis of all the analogies.

Our model performs the best out of all the tested models for positive examples, even if the baselines achieve more than 90% and 80% on Georgian and Spanish, respectively. For the negative examples (invalid analogies) there are three cases: *(i)* our model performs the best (Maltese, Turkish, Japanese), *(ii)* there is no significant difference between our model and the baselines (Arabic, Hungarian), or *(iii)* our model has a lower accuracy but in that case there is no significant difference between the baselines (Finnish, Georgian, German, Navajo, Russian, Spanish). For Russian and Finnish, our model

has an accuracy lower than the baseline by at least 13% for negative examples. However, those baselines are much worse on the positive examples (by at least a 58%).

Overall, our model performs worse on negative than on positive examples, which is coherent with the training process, as we use less than half the number of negative than positive examples during training. Additionally, while our model performs slightly worse than the baselines on most negative examples, it strongly outperforms them for the positive examples.

V. Towards a Transferable Neural Analogy Model?

We applied the model trained on each language to the other languages of the dataset. The objective is to explore the generalization capabilities of the CNN model, and to test its dependence on the training language. In Section V-A, we present experiments where both the classifier and the embedding model were transferred from one language to an other. We ran additional experiments using the embedding model trained on the target language and transferring only the classifier. These results are presented in Section V-B.

A. Embedding Model and Classifier of Another Language than the Target Language

The results for full transfer (*i.e.*, using both the embeddings and the classifier of a language on another language) are presented in Figure 2.

A quick comparison of the results shows that while our deep learning model performs well for base and positive examples, major differences appear for negative examples. This is mostly due to differences in alphabet between the source and target language that cause a large portion of the target alphabet to be unrecognized. This in turn leads to a large amount of 0 embeddings for the characters which tend to be ignored by the model. The high performance on positive and very low performance on negative classification may indicate that, in case of unrecognized characters, the classifier answers "valid analogy" by default. This makes sense as unrecognized characters are treated as padding by the word embedding model and thus should not impact the resulting embedding, and the embedding. The embedding would thus be closer to that of an empty word ε. As $\varepsilon : \varepsilon :: \varepsilon : \varepsilon$ is obvious, a direct consequence of not recognizing characters is the classification as a "valid analogy".

Overall, most languages transfer reasonably well, with an accuracy between 50% to 80%. However, models have an accuracy close to 0% when transferred *from* Georgian, Japanese and Russian or *to* Georgian and Russian, which could be expected as most of the alphabet is not shared. Additionally, models trained on Turkish and Hungarian perform slightly better when compared to those trained on other languages. Interestingly, some models manage to transfer reasonably to Japanese despite the alphabet gap, in particular, from the above mentioned Hungarian and Turkish. Further analysis and experiments would be required to understand what happens in those cases. Also, the quality of the transfer is not symmetric, which means that a model that performs well when transferred from a language A to a language B may perform badly when trained on language B and transferred to language A.

B. Embedding Model of the Target Language but Classifier of Another Language

The results for partial transfer (*i.e.* using the classifier model of a language on another language using the target language's embedding model) are presented in Figure 3.

The results are similar to those of full transfer in that the model transfers well for base and positive examples while differences appear for negative examples. However, we can notice a drop of up to 30% in accuracy for base examples when transferring from Finnish. For negative examples, the performance is above 10% for all but three cases (from Georgian and Spanish to Japanese and from Spanish to Arabic), and above 25% in most cases, which is a clear improvement from full transfer. It is noteworthy that the target of the transfer seems less important than the source language, with Spanish, Georgian and Arabic, as well as Japanese to a lower degree, are languages from which transferred classifiers perform worse.

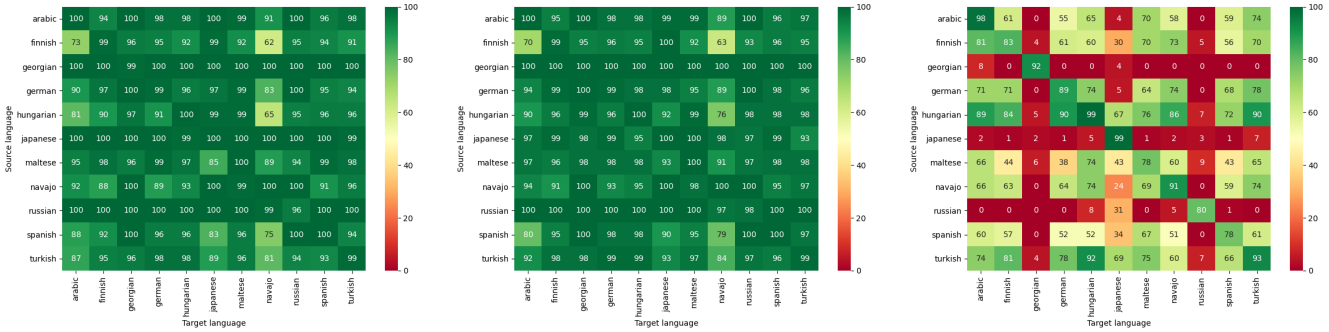
This experiment display the performance of a model where the alphabet gap is not an issue, resulting in a significant improvement of the performance. Nonetheless, the embedding model was not trained together with the classifier which may result in a mismatch in the representation. Additional experiments are required to have a better understanding on what cause the word embedding space to mismatch that of the classifier.

VI. Conclusion

We successfully adapted the approach of [11] from semantic to morphological analogies. In this process, we discussed some of the limitations of the symbolic approaches with regard to analogy applied to morphology related tasks.

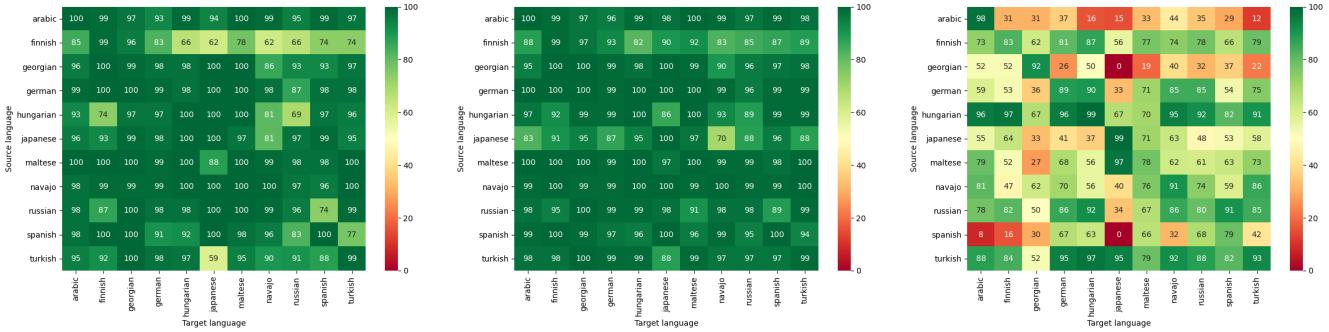
While the semantic approaches mentioned in this paper do not require training, our CNN model is more flexible as it is able to carry over domain and language specificities from the training process. Our model is also more flexible in terms of the properties of analogies that it models. Changing the way the data is augmented will change the way the model behaves, which can allow the adaptation of the set of properties considered for analogies. For example, by dropping uniqueness or central permutation to fit another formalization of analogy. Additionally, our model has strong potential to carry over models of analogy when using an adapted embedding model as we showed with our transfer experiments. This stays true even if the embedding model does not perfectly fit the data.

As we limited ourselves to analogy classification in this paper, further work includes testing the analogy solving aspect (the regression model) of Lim *et al.*'s neural approach on morphology [11]. Also, restricted ourselves to 3 instead of the 24 negative forms per raw analogy used in [11]. Intuitively, using the full set of 24 forms for the negative examples during training will only increase the performance of our



(a) Base (b) Positive (c) Negative

Fig. 2: Accuracy (in %) of fully transferred models (*i.e.* embedding and classifier of a language applied to another language) on Sigmorphon2016, for each of the tree categories of examples.



(a) Base (b) Positive (c) Negative

Fig. 3: Accuracy (in %) of partially transferred models (*i.e.* classifier of a language applied to another language using the target language's embedding model) on Sigmorphon2016, for each of the tree categories of examples.

model, which would lead to outperform the state of the art for negative examples[4]. Furthermore, our early experiments on transferability highlight the potential to transfer and reuse our neural approach across domains. The results also confirm our hypothesis that morphological analogy models are transferable between similar languages (in terms of alphabet and morphology). Further results on transferability could be used to measure morphological similarities between languages. All languages have their own characteristics in terms of the nature of used inflexions (suffixation, prefixation, duplication, etc). We intend to do a thorough study on the impact of these inflexions onto classifying (or solving) analogies, as well as onto the transferability between languages.

Acknowledgments

This research was partially supported by the project "Foundations of Trustworthy AI integrating Learning, Optimisation and Reasoning" (TAILOR) funded by EU Horizon 2020 research and innovation programme under GA No. 952215, and the Inria Project Lab "Hybrid Approaches for Interpretable AI" (HyAIAI).

Experiments were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

[4]This initial intuition has been attested, where better results were achieved for classification task when worked with 8 positive and 8 negative examples.

References

[1] M. Mitchell, "Analogy making as a complex adaptive system," in *Santa Fe Institute Studies in the Sciences of Complexity*. Reading, Mass.; Addison-Wesley; 1998, 2001, pp. 335–360.

[2] ——, "Abstraction and analogy-making in artificial intelligence," *Annals of the New York Academy of Sciences*, 2021.

[3] M. Couceiro, N. Hug, H. Prade, and G. Richard, "Analogy-preserving Functions: A Way to Extend Boolean Samples," in *26th IJCAI*, C. Sierra, Ed., Melbourne, Australia, 2017, pp. 1–7.

[4] F. de Saussure, *Cours de linguistique générale*. Paris: Payot, 1916.

[5] R. Bod, "From exemplar to grammar: A probabilistic analogy-based model of language learning," *Cognitive Science*, vol. 33, no. 5, pp. 752–793, 2009.

[6] H. Behrens, "The role of analogy in language processing and acquisition," *The changing English language: Psycholinguistic perspectives*, pp. 215–239, 2017.

[7] R. Fam and Y. Lepage, "Tools for the production of analogical grids and a resource of n-gram analogical grids in 11 languages," in *11th LREC*, N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds. Miyazaki, Japan: ELRA, 2018.

[8] P.-A. Murena, M. Al-Ghossein, J.-L. Dessalles, and A. Cornuéjols, "Solving analogies on words based on minimal complexity transformation," in *29th IJCAI*, C. Bessiere, Ed., 2020, pp. 1848–1854.

[9] V. Taillandier, L. Wang, and Y. Lepage, "Réseaux de neurones pour la résolution d'analogies entre phrases en traduction automatique par l'exemple," in *6e conférence conjointe (JEP, 33e édition), (TALN, 27e édition), (RÉCITAL, 22e édition). Volume 2 : Traitement Automatique des Langues Naturelles*, C. Benzitoun, C. Braud, L. Huber, D. Langlois, S. Ouni, S. Pogodalla, and S. Schneider, Eds. Nancy, France: ATALA, 2020, pp. 108–121.

[10] H. Prade and G. Richard, "A short introduction to computational trends in analogical reasoning," in *Computational Approaches to Analogical Reasoning: Current Trends*, ser. Studies in Computational Intelligence, H. Prade and G. Richard, Eds. Springer, 2014, vol. 548, pp. 1–22.

[11] S. Lim, H. Prade, and G. Richard, "Solving word analogies: A machine learning perspective," in *15th ECSQARU*, G. Kern-Isberner and Z. Ognjanovic, Eds., vol. 11726. Belgrade, Serbia: Springer, 2019, pp. 238–250.

[12] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *52nd ACL (Volume 2: Short Papers)*. Baltimore, Maryland: ACL, Jun. 2014.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st ICLR, Workshop Track*, Y. Bengio and Y. LeCun, Eds., Scottsdale, Arizona, USA, 2013.

[14] D. Chen, J. C. Peterson, and T. Griffiths, "Evaluating vector-space models of analogy," in *39th CogSci*, G. Gunzelmann, A. Howes, T. Tenbrink, and E. J. Davelaar, Eds., 2017.

[15] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, A. Moschitti, B. Pang, and W. Daelemans, Eds., 2014, pp. 1532–1543.

[16] R. Cotterell, C. Kirov, J. Sylak-Glassman, D. Yarowsky, J. Eisner, and M. Hulden, "The sigmorphon 2016 shared task—morphological reinflection," in *SIGMORPHON, 2016*, M. Elsner and S. Kübler, Eds. Berlin, Germany: ACL, August 2016.

[17] M. Karpinska, B. Li, A. Rogers, and A. Drozd, "Subcharacter information in japanese embeddings: when is it worth it?" in *Workshop on the Relevance of Linguistic Structure in Neural Architectures for NLP*. Melbourne, Australia: ACL, 2018, pp. 28–37.

[18] Y. Lepage, "Analogy and formal languages," vol. 53, 2001, pp. 180–191.

[19] ——, "De l'analogie rendant compte de la commutation en linguistique," Habilitation à diriger des recherches, Université Joseph-Fourier - Grenoble I, May 2003.

[20] L. Miclet, S. Bayoudh, and A. Delhay, "Analogical dissimilarity: Definition, algorithms and two experiments in machine learning," *J. Artif. Intell. Res.*, vol. 32, pp. 793–824, 2008.

[21] M. Couceiro, N. Hug, H. Prade, and G. Richard, "Behavior of Analogical Inference w.r.t. Boolean Functions," in *27th IJCAI*, J. Lang, Ed., Stockholm, Sweden, 2018, pp. 2057–2063.

[22] P. Langlais, F. Yvon, and P. Zweigenbaum, "Improvements in analogical learning: Application to translating multi-terms of the medical domain," in *12th EACL*, A. Lascarides, C. Gardent, and J. Nivre, Eds. Athens, Greece: ACL, 2009, pp. 487–495.

[23] F. Yvon, "Finite-state transducers solving analogies on words," *Rapport GET/ENST&LTCI*, 2003.

[24] N. Chater and P. Vitányi, "Simplicity: a unifying principle in cognitive science?" *Trends in Cognitive Sciences*, vol. 7, no. 1, pp. 19–22, 2003.

[25] A. Cornuéjols and J. Ales-Bianchetti, "Analogy and induction: which (missing) link?" in *Workshop "Advances in Analogy Research: Integration of Theory and Data from Cognitive, Computational and Neural Sciences"*, 1998.

[26] P.-A. Murena, J.-L. Dessalles, and A. Cornuéjols, "A complexity based approach for solving hofstadter's analogies," in *ICCBR (Workshops)*, A. A. Sánchez-Ruiz and A. Kofod-Petersen, Eds., 2017.

[27] A. Drozd, A. Rogers, and S. Matsuoka, "Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen," in *26th COLING*, N. Calzolari, Y. Matsumoto, and R. Prasad, Eds. ACL, 2016, pp. 3519–3530.

[28] C. Vania, "On understanding character-level models for representing morphology," Ph.D. dissertation, University of Edinburgh, 2020.

[29] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT, 2019*, J. Burstein, C. Doran, and T. Solorio, Eds. ACL, 2019, pp. 4171–4186.

[30] Y. Lepage, "Character-position arithmetic for analogy questions between word forms," in *ICCBR 2017 Workshops (CAW, CBRDL, PO-CBR), Doctoral Consortium, and Competitions co-located with the 25th International Conference on Case-Based Reasoning*, A. A. Sánchez-Ruiz and A. Kofod-Petersen, Eds., vol. 2028, Trondheim, Norway, 2017, pp. 23–32.