Workshop Analogies: from Theory to Applications (ATA@ICCBR 2022)

Miguel Couceiro, Esteban Marquer, Pierre-Alexandre Murena, Pierre Monnin

Analogies: from Theory to Applications

Organizers:

Miguel Couceiro (University of Lorraine, CNRS, LORIA)
Esteban Marquer (University of Lorraine, CNRS, LORIA)
Pierre-Alexandre Murena (Aalto University)
Pierre Monnin (Orange)

Program Committee:

Adrien Coulet	Jean Lieber	Henri Prade
Mehdi Kaytoue	Mathieu d´Aquin	Christophe Cerisara
Claire Gardent	Gilles Richard	Laurent Miclet
Steven Schockaert	Yves Lepage	Myriam Bounhas
Sebastien Destercke	Claudia d´Amato	

Analogical proportions, i.e., statements of the form "A is to B as C is to D", are the basis of analogical inference and they are closely related to case-based reasoning and transfer learning. They are have been used on NLP tasks such as automatic machine translation, semantic and morphological tasks, as well as visual question answering with competitive results. Moreover, analogical reasoning can support several machine learning tasks such as classification, decision making, or dataset augmentation. However, other less explored applications could be envisioned such as knowledge discovery and management (e.g., knowledge graphs refinement, data set completion, and alignment), recommender systems, and other AI-related tasks such as explainable AI.

The purpose of this workshop is thus to explore both foundational and applicative aspects of analogical reasoning in various fields, e.g., machine learning, knowledge representation, discovery, and reasoning, as well as in industry practice with real-world data, applications, and associated challenges, for instance, scalability issues.

Sentence Analogies for Text Morphing

Zhicheng Pan[1,*], Xinbo Zhao[1] and Yves Lepage[1]

[1]Waseda University, 2-7 Hibikino, Kitakyushu, 808-0135, Japan

Abstract

Text morphing is a Natural Language Processing (NLP) task which aims at generating sequences of fluent and smooth intermediate sentences between two input sentences, the start and end sentences. In this paper, we show how to use sentence analogies to augment data for this task. We rely on the notion of analogy to produce sequences of sentences exhibiting step-by-step transitions. We use these sequences to fine-tune a large-scale pre-trained language model that is used for text generation. The performance is evaluated by two criteria: fluency and transition smoothness on both the semantic and formal levels. Compared to a variational autoencoder generative model, our model is shown to generate smoother transitions, although the generated sentences are slightly less fluent.

Keywords

Sentence analogy, text morphing, data creation

1. Text Morphing

Text morphing is an NLP task that consists in generating a sequence of sentences that make the transition between a start sentence and an end sentence. Tables 1 and 2 show examples. This acceptation of text morphing is slightly different from the one in [1], where it is nearer to the meaning found in image morphing where information from two or several images is blended into one.

Obviously, text morphing can draw from techniques in Natural Language Generation (NLG) [2]. Traditional methods in NLG generally start from scratch. This is the case of left to right generation using latent sentence vector sampling [3]. In text morphing, we start from two given sentences and generate intermediate sentences. [4] proposes a generative language model for sentences that first samples a prototype sentence from a training corpus and then edits it into a new sentence. Based on that, [5] defined Text Morphing with the goal of generating intermediate sentences that are fluent and smooth between two input sentences.

[3] proposed an RNN-based variational autoencoder generative model which can generate coherent and diverse sentences using the latent space. It can also generate sentences from points between two sentence encodings. The model is called Sentence Variational Autoencoder (Sentence-VAE). Sentence-VAE incorporates distributed latent representations of entire sentences. It uses a continuous latent variable to capture global characteristics. The transitions

ICCBR Analogies'22: Workshop on Analogies: from Theory to Applications at ICCBR-2022, September, 2022, Nancy, France

*Corresponding author.

panzhicheng@toki.waseda.jp (Z. Pan); zhao.symbol@fuji.waseda.jp (X. Zhao); yves.lepage@waseda.jp (Y. Lepage)

Table 1
Example text morphing sequence generated by the Sentence-VAE model (copied from [3]). The start and end sentences are given, the intermediate sentences S_1 to S_4 are generated sentences.

Start sentence	*he was silent for a long moment .*
S_1	*he was silent for a moment .*
S_2	*it was quiet for a moment .*
S_3	*it was dark and cold .*
S_4	*there was a pause .*
End sentence	*it was my turn .*

Table 2
Example text morphing sequence (copied from [5]). The start and end sentences are given, the intermediate sentences S_1, S_2, S_3 are generated sentences.

Start sentence	*The noodles and pork belly was my favourite .*
S_1	*The pork belly was my favourite .*
S_2	*The pork was very good .*
S_3	*The staff was very good .*
S_4	*The staff is very friendly .*
End sentence	*Love how friendly the staff is .*

obtained through variational latent space are smoother and the fluency of the generated sentences is higher. By searching paths through the latent space, it can generate coherent new sentences which interpolate between two already known sentences. Table 1 shows an example of a text morphing sequence generated by Sentence-VAE.

[4] proposed a new generative language model for sentences that first samples a prototype sentence from the training corpus and then edits it into a new sentence [4]. They perform experiments on the Yelp review corpus [6] and the One Billion Word Language Model Benchmark [7]. The result shows that the model they proposed improves the fluency of the generated sentences.

Building on the previous work, [5] took text editing a step further and proposed a novel model called Morphing Networks which can generate intermediate sentences by editing vectors obtained from a start sentence and an end sentence. The generated intermediate sentences should be fluent and the transitions should be smooth. They aim to gradually approach the end sentence by editing the start sentence step by step, that is, with increasing similarity to the end sentence. Each edit produces a new sentence, and ideally, the editing path is smooth because they only change a small part of the sentence, a few words or a phrase, with each edit. Table 2 shows an example that exhibits relatively smooth and natural transition between two sentences.

2. Proposed Method for Text Morphing

In nowadays NLP, it has become classical to fine-tune a large-scale pre-trained language model to perform a given downstream task, as this has been proven to be efficient in many cases. To perform fine-tuning in a supervised way, implies the use of a data set for the task in question. In our case, this means a data set of text morphing sequences.

In this paper, we show how to create text morphing sequences by exploiting the notion of analogy between sentences. This point is the original point in our proposed method.

Our method thus consists of the following two steps. Firstly and most importantly, we use the notion of sentence analogy (Subsection 2.2) to create a data set of text morphing sequences (Subsection 2.1). Secondly, we use this data set to fine-tune a large-scale pre-trained language model (Subsection(3.4) on the task of text morphing.

Below, we detail the original point in our method, i.e., the creation of a data set of text morphing sequences using the notion of analogies between sentences. We also explain how we solve sentence analogies.

2.1. Creating a Data Set of Text Morphing Sequences

We construct a data set of text morphing sequences by solving sequences of analogies between sentences. We start with a sentence analogy $A : B :: C_0 : x,$ where A, B and C_0 are sentences extracted from a data set of sentence analogies and x is unknown.[1] olving the equation delivers a sentence $x = C_1$. We recursively apply the process by replacing C_0 with C_1, etc., leaving A and B unchanged. In this way, we obtain a sequence of sentences C_0, C_1, ..., C_n. It is a text morphing sequence where C_0 and C_n are the start and end sentences and C_1, ..., C_{n-1} are the intermediate sentences.[2] See Figure 1.

Since we are constantly replacing C_{i-1} with the next sentence C_i predicted by sentence analogy, the direction of changes in the entire sequence is given by the direction between A and B. Now, as the variation is, by definition of the analogy $A : B :: C_{i-1} : C_i,$ limited by the variation between A and B, the transitions should be smooth, if A and B are not too distant. The tool used to solve the sentence analogies should be responsible for the fluency of the generated sentences C_i.

To summarize, in this process, the sentence C_0 is transformed slowly step by step into C_n, along the direction defined by A and B. Notice that we give the start sentence C_0, but that we do not know in advance the end sentence C_n.

2.2. Solving Sentence Analogies

The previous process requires a tool to solve sentence analogies. Sentence analogies are more difficult to solve than word analogies (*go* is to $went$ as $walk$ is to $walked$ or $Tokyo$ is to $Japan$ as $Bejing$ is to $China$). The syntactic structure and semantic complexity of sentences makes the difficulty.

[1] http://lepage-lab.ips.waseda.ac.jp/en/projects/kakenhi-kiban-c-18k11447/ See Experimental Results.

[2] In our experiments, we set n to have 1 to 5 intermediate sentences. When n becomes larger, we observe that the same sentence may be repeated in the sequence of sentences.

$$\left.\begin{array}{lllllllll} A & : & B & :: & C_0 & : x \Rightarrow x = & C_1 \\ A & : & B & :: & C_1 & : x \Rightarrow x = & C_2 \\ A & : & B & :: & C_2 & : x \Rightarrow x = & C_3 \\ \vdots & : & \vdots & :: & \vdots & : x \Rightarrow x = & \vdots \\ A & : & B & :: & C_{n-1} & : x \Rightarrow x = & C_n \end{array}\right\} \quad \Longrightarrow \quad (C_0, C_1, C_2, \ldots, C_n)$$

Figure 1: Process of creating a text morphing sequence.

It 's really not that interesting .	:	*It 's really not that hot .*	::	*It 's not that bad .*	: x	$\Rightarrow$ $x =$	*It 's not that cold .*
You 're not from around here , are you ?	:	*You 're not staying here , are you ?*	::	*You 're confused again , are n't you ?*	: x	$\Rightarrow$ $x =$	*You 're disappointed , are n't you ?*

Figure 2: Semantico-formal analogies from the data set released in [10].

In early proposals to solve sentence analogies [8, 9], sentences have been considered as strings of words or characters. The disadvantage is that the semantics of sentences is not controlled. [10] proposed to combine both the form of sentences (strings of words) with the meaning of words (vector representations of words). They released a set of 5,600 so-called semantico-formal analogies in English. Examples are shown in Figure 2.

[11] proposed to learn the mapping between three vector representations of sentences (A, B and C) for and the vector representation of the sentence D solution of the analogy $A : B :: C : D$. The three vectors for A, B and C can be obtained from word or sentence embedding models. To decode the sentence D from its vector representation, they proposed a vec2seq model, implemented as a fully connected network, to map vector representations of sentences onto their corresponding sentences.

Here, we solve sentence analogies using yet another method described elsewhere [12]. It consists in fine-tuning a large-scale pre-trained model on the data set of semantico-formal analogies mentioned above. The fine-tuned model obtained can perform the task of solving sentence analogies directly in an end-to-end manner. Different language models were tested and the most efficient one was a fine-tuned GPT-2 model. We use that one in the experiments reported in this paper.

3. Experiment Settings

3.1. Data Used

The previously mentioned data set of semantico-formal analogies is used to create text morphing sequences that will be used to train a GPT-2 model for the task of text morphing. This data set was created from sentences extracted from the English part of the Tatoeba resource[3].

3.2. Assessment of Text Morphing Sequences

Assessment of text morphing is done according to two dimensions. Firstly, by the smoothness of the transitions between the sentences in the morphing sequences: two consecutive sentences should not differ by too much for the entire sequence to be considered smooth. Secondly, by the quality of each individual intermediate sentence: all sentences generated should sound natural, fluent, grammatical, in a word, it should be reasonable.

3.2.1. Transition Smoothness

We define it as the average of all edit distances between consecutive sentences in the morphing sequence. The edit distance between two strings gives the number of edit operations needed to transform a given string into another one. It is thus particularly well suited for our purpose. Here we use the Levenshtein distance [13] in which deletion, insertion, and substitution are the basic edit operations. Lower scores indicate smoother transitions.

3.2.2. Fluency of a Text Morphing Sequence

Perplexity, as classically used in language modelling, is a measure of the reasonableness of sentences. We thus define the fluency of a text morphing sequence as the average of the perplexity scores over all intermediate sentences (excluding the start and end sentences). A lower score indicates higher fluency.[4]

3.3. Creation of a Data Set of Text Morphing Sequences Using Sentence Analogies

As mentioned at the end of Section 2.2, to solve sentence analogies, we fine-tune a pre-trained language model, GPT-2 [14], on the task of solving sentence analogies. The sentence analogies used during this training are from the semantico-formal analogy data set mentioned in Section 2.2.

To create a data set of text morphing sequences, we then use each semantico-formal sentence analogy as a starting point as described in Section 2.1 and illustrated in Figure 1.

We asses the quality of the created text morphing sequences with the metrics introduced in Section 3.2, but, in addition, we compare with an existing model.

[3] https://tatoeba.org/

[4] We use the a 3-gram language model, trained on the Tatoeba corpus, with the KenLM toolkit https://github.com/kpu/kenlm to compute the perplexity of the sentences.

Table 3
The GPT-2 fine-tuning settings for text morphing.

Hyperparameter	Value
GPT-2 model	345M
Optimizer	adam
Batch size	1
Learning rate (LR=2)	0.00002

[5] did not release their code, although they claim better results than the Sentence-VAE model [3]. The code for this latter model is available[5]. So we adopt it as our baseline. In our experiments, we trained the Sentence-VAE model using the Tatoeba data set from which the above-mentioned semantico-formal analogies were extracted. By exploring the paths between the start and end sentences created with our method, in the latent space of the obtained Sentence-VAE model, we can generate a certain number of coherent sentences, which constitute a text morphing sequence.

In this way, we can compare the transition smoothness and the fluency of two comparable sets of text morphing sequences, created from the same start and end sentences, by two methods, the Sentence-VAE model, and our proposed method.

3.4. Fine-Tuning a Pre-Trained Model with the Data Set of Text Morphing Sequences Created Using Sentence Analogies

We fine-tune the pre-trained GPT-2 model using the sentence sequences generated in the previous section. For comparison, as in the previous section, we still use the Sentence-VAE model as a baseline model. Due to limitations in memory, we choose the medium-sized GPT-2 model (345M). The GPT-2 fine-tuning parameters are shown in Table 3. For the baseline model, we trained the Sentence-VAE model using the Tatoeba corpus dataset which consists of 110,000 English sentences.

GPT-2 [14] is a large transformer-based language model created by OpenAI. GPT-2 uses the Decoder structure of the Transformer [15], with some changes to the Transformer Decoder. They verified that unsupervised language modeling is able to learn the features required for supervised tasks. GPT-2 pretraining uses the foregoing to predict the next word, which is suitable for text generation tasks since text generation usually generates the next word based on currently available information.

GPT-2 is a large model based on transformer training on a very large dataset with a large scale, and GPT-2 has a good performance in text generation, both in terms of contextual coherence and sentiment expression.

[5] https://github.com/timbmg/Sentence-VAE

Table 4
Quality of text morphing sequences. On the *left*, in the creation of the data set to be used in fine-tuning. On the *right*, in performing the task of text morphing with the fine-tuned model. For both measures of transition smoothness and fluency, the lower, the better.

	Data Set Creation		Text Morphing	
	Transition Smoothness	Fluency	Transition Smoothness	Fluency
Sentence-VAE [3]	3.52	**1.35**	3.73	**1.36**
Our method	**1.31**	1.57	**0.72**	1.75

Table 5
Statistics of created morphing sequences dataset.

	sequences	sentences	words/sent	chars/sent
generated dataset	5,228	26,140	6.41	23.82

4. Results

4.1. Results for the Creation of the Data Set of Text Morphing Sequences

The quality of the created text morphing sequences, that will be used afterwards to train a large-scale language model for the task of text morphing, is shown in Table 4. Our proposed approach delivers smoother sentences which are semantically relatively correct, in comparison with the Sentence-VAE model proposed in [3] for generating morphing sequences.

Table 4 shows that the transition smoothness (average of edit distance between consecutive sentences) of Sentence-VAE is 3.52, while it is 1.31 with our proposed method. This means that for each transition, the Sentence-VAE model changes on average three and a half words on average, while our proposed method changes 1.3 words only, less than half in comparison. The average number of words per sentence being 6.7, the baseline method changes half the sentence at each transition. Our method makes more subtle and smoother changes.

The fluency, as measured by perplexity, is 1.35 in the method using Sentence-VAE, while it is 1.57 in our method (the scores are small because the sentences are short). According to these numbers, the sentences generated by the Sentence-VAE model are more reasonable, but whether there is a real difference may be disputable. We conclude that, in comparison with the Sentence-VAE model, our proposed method delivers smoother sentences that are relatively fluent.

The following Table 5 shows basic statistics of our created dataset. An example of generated morphing sentences is given in Table 6 below.

Table 6
Morphing sequences obtained with the Sentence-VAE method (on the *left*) and our proposed method (on the *right*) for the same start and end sentences.

Sentence-VAE [3]	Our method [this paper]
I 'm ready to go .	*I 'm ready to go .*
what do you think of this is ? *this is a good textbook .* *i have a lot of money in this store .*	*I really have to go .* *I really need to go .* *I need to go somewhere.*
I have to go somewhere .	*I have to go somewhere .*

4.2. Results for the Text Morphing Task

The quality of text morphing is shown in the same table as before, Table 4. The results of this experiment are similar to those obtained in the previous section when creating a data set of text morphing sequences. This indicates that our trained model can deliver smoother sentences which are semantically relatively correct, in comparison with the Sentence-VAE model for the task of text morphing.

The transition smoothness of the Sentence-VAE model is 3.73, while it is 0.72 with our proposed method. The previous remarks made above apply similarly here for this model. It is not a surprise as we use it here in the same way as before. Our proposed method shows improvement in transition smoothness relatively to the creation of text morphing sequences: the average edit distance between two consecutive sentences has been almost divided by two.

The perplexity of the method using the Sentence-VAE model is 1.36, while the perplexity with our proposed method is 1.75. Again, there is no difference between the scores in the data creation step and the text morphing tesk for the Sentence-VAE model because we use it in the same way in both cases. Our proposed method generates sentences with a slightly worse perplexity in the text morphing task compared with the creation of text morphing sequences using sentence analogy. However, again, we can conclude that our proposed fine-tuned model delivers sentences which are relatively fluent, but smoother, in comparison with the Sentence-VAE model.

4.3. Discussion

When creating text morphing sequences, we observed that, sometimes the same sentences were generated repeatedly or several sentences were generated alternately. We explain these phenomena by the relative shortness of the sentences used. The sentences contained in our data set are less than 10 words long. Shorter sentences allow for fewer options for changes when the text is morphed, and sometimes repetition occurs, which induces no change.

Table 7
Examples of text morphing sequences (of length 3) generated by the fine-tuned GPT-2 model. Start sentences on the first row, end sentences on the last row.

I deserve this .	*I really do not know .*	*I see the problem .*
I do not deserve this .	*I do not know .*	*I know the truth .*
I deserve that .	*I do not know anything .*	*I know the problem .*
I do not need that .	*I do not know .*	*I know the truth .*
I do not need a girlfriend .	*I do not understand anything .*	*I know the solution .*

5. Conclusion

We proposed to perform text morphing by fine-tuning a large-scale pre-trained language model on the task, as is classical nowadays in NLP. But for that, data was needed. We relied on analogies to create text morphing sequences. We proposed an original method which consists in starting with an analogical equation and in letting the solver perform changes in the direction defined by the two terms on the left of the analogical equation. Variations are obtained step by step and this results in text morphing sequences.

The performance of the fine-tuned model was evaluated with transition smoothness and fluency. Our model achieved more than three times smoother transitions than the baseline we considered, the Sentence Variational Autoencoder generative model. However, the baseline was shown to generate slightlymore fluent sentences than our proposed model.

References

[1] R. A. Connor, Multi-stage text morphing, patent US 2011/0184725 A1, 2011. URL: https://patents.google.com/patent/US20110184725.

[2] A. Gatt, E. Krahmer, Survey of the state of the art in natural language generation: Core tasks, applications and evaluation, Journal of Artificial Intelligence Research 61 (2018) 65–170.

[3] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, S. Bengio, Generating sentences from a continuous space, in: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, (CoNLL2016), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 10–21. URL: https://www.aclweb.org/anthology/K16-1002. doi:10.18653/v1/K16-1002.

[4] K. Guu, T. B. Hashimoto, Y. Oren, P. Liang, Generating sentences by editing prototypes, Transactions of the Association for Computational Linguistics 6 (2018) 437–450. URL: https://www.aclweb.org/anthology/Q18-1031. doi:10.1162/tacl_a_00030.

[5] S. Huang, Y. Wu, F. Wei, M. Zhou, Text morphing, ArXiv (not published elsewhere) abs/1810.00341 (2018).

[6] N. Asghar, Yelp dataset challenge: Review rating prediction, 2016. URL: https://arxiv.org/abs/1605.05362.

[7] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, One billion word benchmark for measuring progress in statistical language modeling, CoRR abs/1312.3005 (2013).

[8] M. Nagao, A framework of a mechanical translation between Japanese and English by analogy principle, in: A. Elithorn, R. Banerji (Eds.), Proceedings of the international NATO symposium on Artificial and human intelligence, Elsevier Science Publishers, NATO, 1984, pp. 173–180. URL: http://www.mt-archive.info/Nagao-1984.pdf.

[9] Y. Lepage, G. Peralta, Using paradigm tables to generate new utterances similar to those existing in linguistic resources, in: Proceedings of the 4th internation conference on Language Resources and Evaluation (LREC 2004), volume 1, Lisbon, 2004, pp. 243–246.

[10] Y. Lepage, Semantico-formal resolution of analogies between sentences, in: Z. Vetulani, P. Paroubek (Eds.), Proceedings of the 9th Language & Technology Conference (LTC 2019) – Human Language Technologies as a Challenge for Computer Science and Linguistics, 2019, pp. 57–61. URL: http://lepage-lab.ips.waseda.ac.jp/media/filer_public/32/04/32049346-75dd-4bd1-93cc-ae221e49a2e9/ltc-005-lepage.pdf.

[11] L. Wang, Y. Lepage, Vector-to-sequence models for sentence analogies, in: IEEE (Ed.), Proceedings of the 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS 2020), 2020, pp. 441–446. URL: https://ieeexplore.ieee.org/document/9263191. doi:10.1109/ICACSIS51025.2020.9263191.

[12] L. Wang, Z. Pan, H. Xiao, Y. Lepage, Solving sentence analogies by using embedding models combined with a vector-to-sequence decoder or by fine-tuning pre-trained language models, 2022. Under review.

[13] V. Levenshtein, Binary codes capable of correcting deletions, insertions and reversals, Soviet Physics-doklady 10 (1966) 707–710.

[14] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language Models are Unsupervised Multitask Learners, Technical Report, OpenIA, 2019.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems (NIPS 2017), volume 30, 2017, pp. 6000–6010.

Transferring Learned Models of Morphological Analogy

Esteban Marquer[1,*], Pierre-Alexandre Murena[2] and Miguel Couceiro[1]

[1]Université de Lorraine, CNRS, LORIA, F-54000, France
[2]HIIT, Aalto University, Helsinki, Finland

Abstract

Analogical proportions are statements of the form "A is to B as C is to D", which have been extensively studied in morphology. Recent advances on learning models of analogy from quadruples pave the way for data-driven modeling and analysis of analogy. In morphology, recent work introduces a neural network classifier for morphological analogies (ANNc). In this paper, we study the transferability of ANNc across different axiomatic settings to show the importance of the data augmentation in the modeling of analogy. We also provide experimental results on transfer between two morphology datasets (Sigmorphon2016 and Sigmorphon2019) and between more than 27 languages to draw parallels between transfer performance and proximity between language families.

Keywords

Transfer, Morphological analogies, Analogy detection

1. Motivation and Context

The past decade has seen an increasing interest in analogical reasoning (AR) and of analogical proportions (APs), which are statements that four elements A, B, C, D are in analogy (usually written $A : B :: C : D$). Indeed, AR and APs are useful not only in the study of the mechanisms of human cognition [1] but also for applications in artificial intelligence [2, 3]. There are two basic tasks associated with AR: the first is *analogy detection* that corresponds to the task of deciding whether a quadruple A, B, C, D constitutes a valid AP, and the second is *analogy solving* that corresponds to finding the solution of an *analogical equation*, *i.e.*, an AP $A : B :: C : X$ where X is unknown.

Analogies between words and strings of symbols has long been studied [4, 5, 6, 7, 8, 9, 10] and makes for an experimental setting in which a wide range of analogies appear, from simple ($a : aa :: b : bb$) to more complex (*word*:*language*::*note*:*music*). In this paper, we focus on the study of morphological analogies, *i.e.*, analogies modeling changes of morphemes. In particular, we study how the deep learning approach for detecting morphological analogies proposed in [11, 12] behaves when transferred across domains. We call this approach Analogy Neural

ICCBR Analogies'22: Workshop on Analogies: from Theory to Applications at ICCBR-2022, September, 2022, Nancy, France

*Corresponding author.
esteban.marquer@loria.fr (E. Marquer); pierre-alexandre.murena@aalto.fi (P. Murena); miguel.couceiro@loria.fr (M. Couceiro)
0000-0003-2315-7732 (E. Marquer); 0000-0003-4586-9511 (P. Murena); 0000-0003-2316-7623 (M. Couceiro)

Network classifier (ANNc). Note that morphological transformations are linked to changes in the syntactic role of a word.

1.1. Axiomatic Setting

The notion of analogy is not consensual and there have been several efforts to create a common logical framework for AR that follow different axiomatic and logical approaches [6, 8]. For instance, Lepage [7] introduces the following 4 axioms in the linguistic context for analogical proportions: *symmetry* (if $A : B :: C : D$, then $C : D :: A : B$), *central permutation* (if $A : B :: C : D$, then $A : C :: B : D$), *strong inner reflexivity* (if $A : A :: C : D$, then $D = C$), and *strong reflexivity* (if $A : B :: A : D$, then $D = B$). While these axioms seem reasonable in the word domain, they can be criticized in other application domains [13]. A functional view of analogy is to consider a transformation f such that $B = f(A)$ and $D = f(C)$, resulting in analogies of the form $A : f(A) :: C : f(C)$ [5, 14]. As such, A and C may not be in the same conceptual domain as B and D. For example, A, C can be cities and B, D countries: *Marseille*:*France*::*Lyon*:*France* is an acceptable analogy (both Marseille and Lyon are cities of France). This is also an example where central permutation can be problematic, as it would give *Marseille*:*Lyon*::*France*:*France* which implies that Lyon and Marseille are identical due to strong inner reflexivity.

In this work, we use model transfer to compare how ANNc behaves when different sets of axioms are considered for training and evaluation. In particular, we consider axiomatic settings in which central permutation is accounted for differently, that we detail in Subsec. 2.2. The experimental results of this comparison are reported in Sec. 3.

1.2. Previous Work on Analogy Detection

The analogy detection task corresponds to classifying quadruples A, B, C, D into valid or invalid analogies. In other words, it can be seen as a binary classification task. Morphological analogy detection is used in the context of analogical grids [15], *i.e.*, matrices of transformations of various words, similar to paradigm tables in linguistics [16]. To detect analogies and build the analogical grids, Fam and Lepage [15] use the number of characters occurrences and the length of the longest common subword.

In the context of semantic word analogies, Bayoudh *et al.* [4] use Kolmogorov complexity as a distance measure between words for analogy detection, and Lim *et al.* [17] implement a data-driven approach. Using a dataset of semantic analogies, Lim *et al.* learn a neural network to classify quadruples A, B, C, D into valid or invalid analogies, using their embeddings e_A, e_B, e_C, and e_D. We adapt the latter approach to morphology by replacing the original GloVe [18] semantic embedding model by a character-level embedding model in our previous work [11], which significantly outperforms previous approaches.

Analogy detection and solving are closely related tasks. For instance, the morphological analogy solving approach of Langlais *et al.* [19] and the one of Murena *et al.* [14] are used as analogy classifiers in [11]: given a quadruple A, B, C, D, if D is in the predicted solutions of $A : B :: C : X$ then $A : B :: C : D$ is a valid analogy, otherwise it is invalid. Other works on solving analogies on character strings can be found in the literature, including Copycat

by Hofstadter and Mitchel [5] but also works relying on embedding spaces [20, 17, 21, 9, 10]. For instance, the work of Lim *et al.* [17] also proposes a model for analogy solving in addition to ANNc. The former model was adapted to morphological word analogies in our previous work [21] and outperforms generative methods that do not rely on deep learning.

1.3. Model Transfer in Machine Learning

In this article, by transfer we mean applying a machine learning model on a *target* domain, different from the *source* domain used to train/learn the model. Different types of transfer are possible, from directly applying a model to the target model, to transferring the model and *finetuning* it on the target domain. The latter method is a type of *model adaptation*, which consists in altering a transferred model to fit the new data and which is a key step in the transfer methodology. It is also possible to transfer a part of a model and reuse it as a component of a larger model, as is usually done with large pretrained embedding models such as BERT [22], wav2vec2 [23], or vision transformers [24].

Transferring a model can serve two main purposes: achieving satisfying performance on the target domain while dealing with issues of the target domain or the model (lack of data or of labeled data, large training time, biases, *etc.*), or studying differences in the behavior of the model on different domains. In this work, we focus on the latter aspect.

1.4. Previous Experiments on Transferring ANNc

In [12], we performed multiple transfer experiments with ANNc on analogies extracted from Sigmorphon2016 [25] and Japanese Bigger Analogy Test Set [26] (which are now available in Siganalogies [27]). We transfered between languages to explore how the analogy model could generalize between domains, and built models on a subset of representative languages to determine the feasibility of a more general model of analogy. In both settings we obtained encouraging performance, but we were not able to fully explain the difference in performance between the languages used. We first experimented with what we called *full transfer*, in which all the components of the approach (character encoder, morphological embedding model and ANNc) are trained on the source language and transferred to the target language, without finetuning. This approach produced good overall results except on some languages using non-roman characters, and is the approach we take for the present article. To solve this *alphabet gap* issue, we used *partial transfer*, *i.e.*, the character encoder and morphological embedding model trained on the target language are reused instead of the ones trained on the source language. While this approach improved performance in case of *alphabet gap*, it was still far from the performance of models trained on the target language, probably due to a mismatch between the embedding model and the embedding space used by ANNc.

1.5. Our Contribution

In this paper, we introduce general elements of our experimental setting in Sec. 2. We extend the results of [12] in several ways:

- in Sec. 3, we use transfer to determine the impact of the axiomatic setting on the performance of the model, as mentioned above, and confirm that different training procedures results in compliance to different sets of axioms;
- in Sec. 4, we confirm that ANNc coupled with the morphological embedding of [11] generalizes to similar data, by transferring models of analogies in 8 languages between the Sigmorphon2016 and Sigmorphon2019 dataset;
- in Sec. 5, we leverage 42 high resource languages of Sigmorphon2019 [28] to extend previous results on inter-language transferability, and confirm previous hypotheses on the *alphabet gap* issue and the transferability of morphological analogies between related languages.

2. Datasets, Axiomatic Setting and Model Transfer

In this section we first present the analogical data. Then, we detail the default axiomatic setting CP (accepting central permutation as an axiom) and two variants $\overline{CP}$ (explicitly refusing central permutation) and $\neg CP$ (not taking central permutation into account). Finally, we specify our training, evaluation and transfer protocol.

2.1. Datasets

In our experiments we use the analogies available in Siganalogies [27], which are extracted from three datasets: Sigmorphon2016 [25], Japanese Bigger Analogy Test Set [26], and Sigmorphon2019 [28]. In Siganalogies the analogies are obtained by associating four words A, B, C, D, with B is a morphological transformation of A (*i.e.*, $B = f(A)$) and similarly $D = f(D)$, such that the morphological transformations are identical.

2.2. Data Augmentation and Axiomatic Setting

In previous works on ANNc, the model was trained using training examples obtained by permuting the four words of each analogy in the dataset. For the positive class, permutations of four words resulting in *valid* analogies (P^+) are generated from each analogy $A : B :: C : D$ in the dataset. For the negative class, permutations resulting in *invalid* analogies (P^-) are generated from each analogy of P^+, as they all are valid analogies.

In introduction, we mention the possibility of using different axiomatic settings, which leads us to experiment with *central permutation* among the most discussed axioms of analogical proportions [13] in Sec. 3. For this purpose, we consider three axiomatic settings (CP, $\neg CP$, and $\overline{CP}$) described below.

In the setting of [11, 12, 17], that we call CP, the axioms of [7] are used and central permutation is considered as an axiom for APs. Central permutation is thus used to generate the permutations in P^+_{CP}. In particular, the permutations in P^+_{CP} (Eq. (1)) can be obtained by applying successively *central permutation* and *symmetry*. Permutations that contradict these two axioms or *strong inner reflexivity* are used to obtain P^-_{CP}. Given a base form $A : B :: C : D$, P^+_{CP} and P^-_{CP} are as follows:

$$P^+_{CP} = \{\langle A, B, C, D\rangle, \langle C, D, A, B\rangle, \langle B, A, D, C\rangle, \langle D, C, B, A\rangle,$$

$$\langle A, C, B, D\rangle, \langle C, A, D, B\rangle, \langle B, D, A, C\rangle, \langle D, B, C, A\rangle\} \tag{1}$$

$$P^-_{CP} = \bigcup_{\langle A', B', C', D'\rangle \in P^+_{CP}} \{\langle A', A', C', D'\rangle, \langle B', A', C', D'\rangle, \langle C', B', A', D'\rangle\} \tag{2}$$

We consider two settings in which central permutation is not an axiom: $\neg CP$ in which we discard the central permutation axiom, and $\overline{CP}$ in which we explicitly consider that applications of central permutation are invalid analogies. Discarding central permutation to obtain $\neg CP$ is the simplest way to refute the central permutation axiom, and results in the following sets of permutations:

$$P^+_{\neg CP} = \{\langle A, B, C, D\rangle, \langle C, D, A, B\rangle, \langle B, A, D, C\rangle, \langle D, C, B, A\rangle\} \tag{3}$$

$$P^-_{\neg CP} = \bigcup_{\langle A', B', C', D'\rangle \in P^+_{\neg CP}} \{\langle A', A', C', D'\rangle, \langle B', A', C', D'\rangle\} \tag{4}$$

For $\overline{CP}$ we go one step further in refuting the central permutation axiom by considering that applications of central permutation are invalid analogies, as mentioned above. To do so, central permutation is removed from the valid permutations (as in $P^+_{\neg CP}$) and added to the permutations of the invalid class:

$$P^+_{\overline{CP}} = P^+_{\neg CP} \tag{5}$$

$$P^-_{\overline{CP}} = \bigcup_{\langle A', B', C', D'\rangle \in P^+_{\overline{CP}}} \{\langle A', A', C', D'\rangle, \langle B', A', C', D'\rangle, \langle B', A', C', D'\rangle, \langle A', C', B', D'\rangle\} \tag{6}$$

In [11, 12] a subset of 8 permutations from P^-_{CP} is randomly sampled during training to obtain balanced classes (8 permutations of P^-_{CP} for 8 permutations of P^+_{CP}). For $\overline{CP}$ and $\neg CP$, to obtain balanced classes with a number of permutations comparable to CP, 8 permutations from each class are randomly sampled[1] during training.

2.3. Model Training and Evaluation and Transfer Method

The training of ANNc on the source domain is done in the same setting as [11], *i.e.*, using 5×10^4 analogies for the dataset and, for each, generating permutations as mentioned above to obtain 8 samples of the valid and 8 of the invalid class. Similarly, the testing is done on 5×10^4 base analogies and all the corresponding permutations. To evaluate the model, we use the *balanced accuracy*, *i.e.*, the average of the accuracy of the valid class and the accuracy of the invalid class, thus ignoring the number of permutations seen in each class. For the experiment on the axiomatic setting (Sec. 3) and on the transfer between datasets (Sec. 4) we use 10 random seeds to ensure stability across random initialization and random selection of the base analogies. For the experiment on the transfer between languages (Sec. 5), due to the large number experiments to perform, a single random seed is used.

In our experiments, the classifier and the embedding model with the character vocabulary of the source domain are transferred to the target domain, and the evaluation is performed without finetuning the models on the target domain. For simplicity, we write $source \rightarrow target$ to denote the transfer from the source domain to the target domain.

[1] If $n > 8$ permutations are available for the class, 8 different permutations are randomly selected. If $n < 8$ permutations are available, $8 - n$ randomly selected permutations are added, ensuring that each permutation appears at least once.

3. Impact of the Axiomatic Setting on the Classification Performance

The purpose of our first experiment is to study the impact of the training procedure of ANNc on the permutations it considers valid or invalid. To confirm that using different training procedures results in models that fit different axiomatic settings, we compare how models trained in the three settings described in Subsec. 2.2 (CP, $\neg CP$, and $\overline{CP}$) behave when transferred to the other settings.

3.1. Experimental Setup

We use the 11 languages of [11], *i.e.*, Sigmorphon2016 and Japanese Bigger Analogy Test Set, and transfer between the three axiomatic settings described in Subsec. 2.2: CP (using P^+_{CP} and P^-_{CP}), $\neg CP$ (using $P^+_{\neg CP}$ and $P^-_{\neg CP}$), and $\overline{CP}$ (using $P^+_{\overline{CP}}$ and $P^-_{\overline{CP}}$). Intuitively, the expected behavior is the following, also represented in the top left corner of Fig. 1:

- each model is expected to perform best on when the source setting is the same as the target setting ($CP \to CP$, $\neg CP \to \neg CP$, and $\overline{CP} \to \overline{CP}$);
- both $CP \to \overline{CP}$ and $\overline{CP} \to CP$ are expected to perform poorly, as the source and target settings are incompatible *w.r.t* CP;
- both CP and $\overline{CP}$ are expected to perform well on $\neg CP$, as the permutations in $\neg CP$ are common to both CP and $\overline{CP}$;
- the performances of $\neg CP \to CP$ and $\neg CP \to \overline{CP}$ are hard to predict, as $\neg CP$ has no constraints *w.r.t* CP.

3.2. Results and Discussion

The results of the experiment are reported in Fig. 1, and for all languages we observe the expected results with minor variations. First, on the target $\neg CP$ setting, all the models perform equally, instead of $\neg CP \to \neg CP$ performing slightly better. Second, the performance of $CP \to \overline{CP}$ and $\overline{CP} \to CP$ are not as low as expected, with the peculiarity that $CP \to \overline{CP}$ always outperforms $\overline{CP} \to CP$ by roughly 10%.

On the one hand, these results confirm that the training procedure does have an impact on which permutations will be considered valid or invalid by the model. On the other hand, the observed results match the expected Z shape for all languages, which supports the intuitions used to construct the expected results. As these intuitions rely on the differences between the axiomatic settings, this experiment confirm that models of a specific axiomatic setting can be obtained with the corresponding training procedure.

4. Transfer Between Morphological Datasets

The purpose of our second experiment is to check whether the model is able to generalize to a closely related domain with a slightly different distribution of morphological transformations. Indeed, for each language the morphological transformations in Sigmorphon2016 [25] and

Figure 1: Balanced accuracy of, 10 per training setting. In the top left corner, a representation of the expected results

Sigmorphon2019 [28] are not exactly the same but the morphology of the language does not change.

4.1. Experimental Setup

For each language present in both Sigmorphon2016 and the high resource languages of Sigmorphon2019, we consider two transfer directions and two baselines:

- **19** $\rightarrow$ **16**: the transfer from the 2019 to the 2016 version of the language;
- **16** the baseline for 19 $\rightarrow$ 16: model trained and tested on Sigmorphon2016;
- **16** $\rightarrow$ **19**: the transfer from the 2016 to the 2019 version of the language;
- **19** the baseline for 16 $\rightarrow$ 19: model trained and tested on Sigmorphon2019.

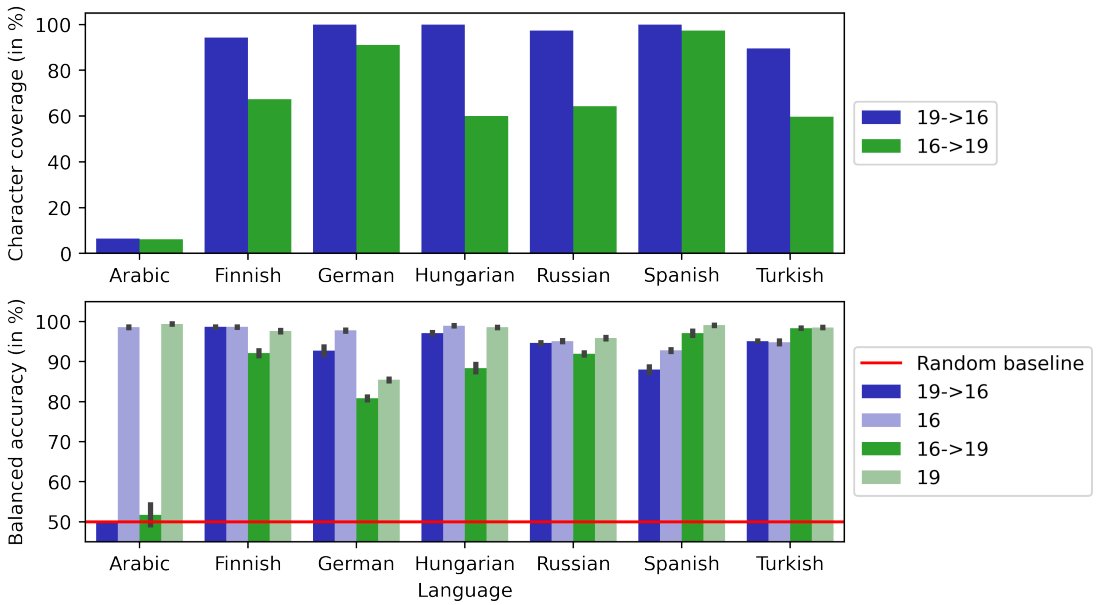

Figure 2: On the top: percentage of coverage of the target language characters by the source language characters. On the bottom: percentage of accuracy of the transferred model compared to the model trained on the target setting.

4.2. Results and Discussion

As shown in Fig. 2, the performance of the transferred model is comparable to or slightly lower than the non-transferred model. A significant correlation is noticeable between the performance of the transferred model and the character coverage of the target domains by the source domains (*i.e.*, the number of characters present in both domains divided by the number of characters present in the target domain), with a Pearson correlation coefficient of $r = 0.9639$ for $19 \rightarrow 16$ and $r = 0.7595$ for $16 \rightarrow 19$. When normalizing the transfer performance by the performance trained on the target domain, the correlation goes up to $r = 0.9739$ for $\frac{19 \rightarrow 16}{16}$ and $r = 0.8639$ for $\frac{16 \rightarrow 19}{19}$. A critical case of this correlation can be seen for Arabic, which is romanized in Sigmorphon2016 and not Sigmorphon2019, leading to a coverage close to 0%.

These results identify character coverage as a key factor in the transfer performance between strongly related domains. In this setup, the embedding model is the main source of performance loss.

5. Transfer Between Languages

To go beyond the limitations of character coverage, our third experiment leverages the large amount of multilingual data available in Sigmorphon2019 [28] to experiment following a similar intuition as in [12] but only between languages with similar alphabets.

5.1. Experimental Setup

We experiment with transfer between the high resource languages of Sigmorphon2019. We exclude Basque and Uzbek as they have less than 5×10^4 analogies. From the results of Sec. 4, we know that the amount of characters in common between the source and target domains strongly impacts the transfer performance. We extract clusters of languages sharing a significant part of their alphabets, and we transfer only within each cluster. This allows us to omit transfers likely to perform poorly due to the alphabet gap. We perform a total of 740 transfers, excluding cases where the source and target are the same. Compared to our other experiments, we reduce the number of test analogies from 5×10^4 to 5×10^3 and use a single random seed.

We use hierarchical clustering with the nearest point algorithm to get the clusters. Instead of the coverage between the source and the target language which is asymmetric, we use the Jaccard index[2] between the alphabets of the languages as a similarity measure. Using a threshold of 40% on the Jaccard and excluding singletons, we extract four clusters of at least two elements[3]. Based on our observations on the coverage[4], we consider relevant to include Romanian in both the Roman and Cyrilic clusters, and obtain the following language clusters named after the dominant alphabetic setting:

1. **Roman** cluster: Albanian, Asturian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Hungarian, Irish, Italian, Kurmanji, Latin, Latvian, Polish, Portuguese, Romanian, Slovak, Slovene, Sorani, Spanish, Swahili, Turkish, Welsh, and Zulu;
2. **Cyrillic** cluster: Adyghe, Bashkir, Belarusian, Bulgarian, Romanian, and Russian;
3. **Arabic** cluster: Arabic, Persian, and Urdu;
4. **Devanagari** cluster: Hindi and Sanskrit.

5.2. Results and Discussion

Once the languages with an overlap lower than 40% are eliminated, the Pearson correlation between the performance and the character coverage drops to $r = 0.6565$ for clusters 2, 3 and 4. For cluster 1, the largest cluster, coverage and performance appear uncorrelated with $r = 0.0379$. Similar values are observed when normalizing the transfer performance by the performance trained on the target domain. We report transfer performance for cluster 1 in Fig. 3.

We do not exclude that these correlations are influenced by the smaller amount of data used (only one seed, fewer testing analogies than usual). However, such a significant drop in correlation is unlikely if only this bias is involved. In fact, the tendencies we observe in the performance matrix indicate that the performance is linked to the language being used as a source language (*e.g.*, horizontal bar for English), and to the one being used as a target language (*e.g.*, vertical bars for Asturian and German). This behavior is likely due to either the quality of the learned model (how well it performs in general) or to the morphological similarities of some languages (at least within Sigmorphon2019). We exclude the former hypothesis, as only

[2]The Jaccard index between two finite sets A and B is $J(A, B) = \frac{A \cap B}{A \cup B}$.
[3]The corresponding dendrogram is provided in Appendix Fig. 2
[4]We provide the matrix of coverage for Sigmorphon2019 in appendix (Fig. 1).

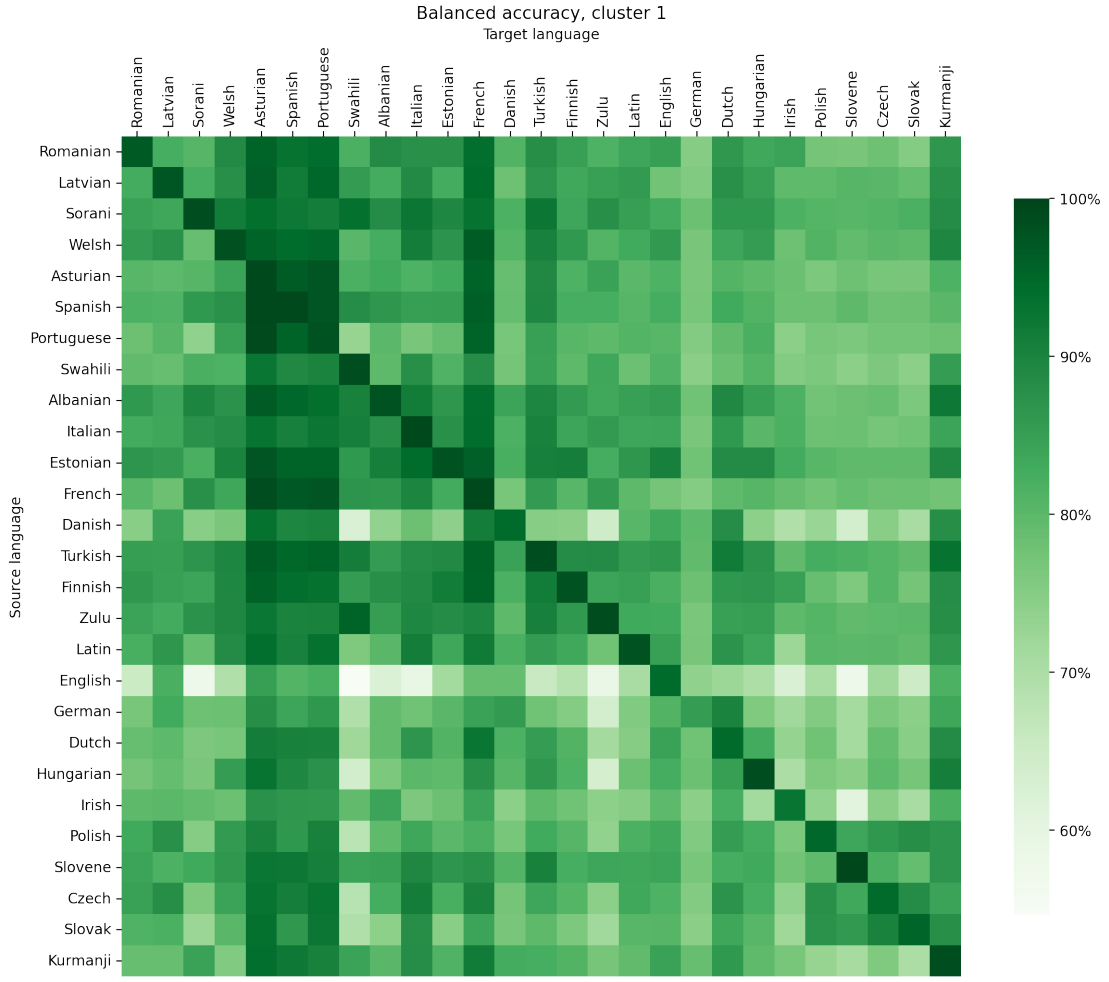

Figure 3: Transfer accuracy within cluster 1.

tendencies in the behavior as a source language (*i.e.*, horizontal bars) would be observed, while we mostly observe tendencies in the behavior as a target language (*i.e.*, vertical bars).

To confirm the influence of language similarities on performance, we explore hierarchical clustering within cluster 1 to study which key groups appear. When considering the behavior of the language as a target domain (*i.e.*, using performance from different source languages as a features for the clustering) rather than as a source domain, the clusters are more distinct. We focus on clusters extracted from the former, which can be seen in the dendrogram in Fig. 4. As a first analysis, we compare the corresponding clusters with language families as defined in Wikipedia. The Wikipedia page of each language contains a box with key information, the *infobox*. We use the "Language family" field of the infobox in the page of each language to determine how closely related they are, after minor corrections. The tree structure in Appendix Fig. 3, summarizes this information, with the leaves colored to match the colors of the clusters on Fig. 4. We find that the small clusters, which are the most easily distinguishable by the clustering

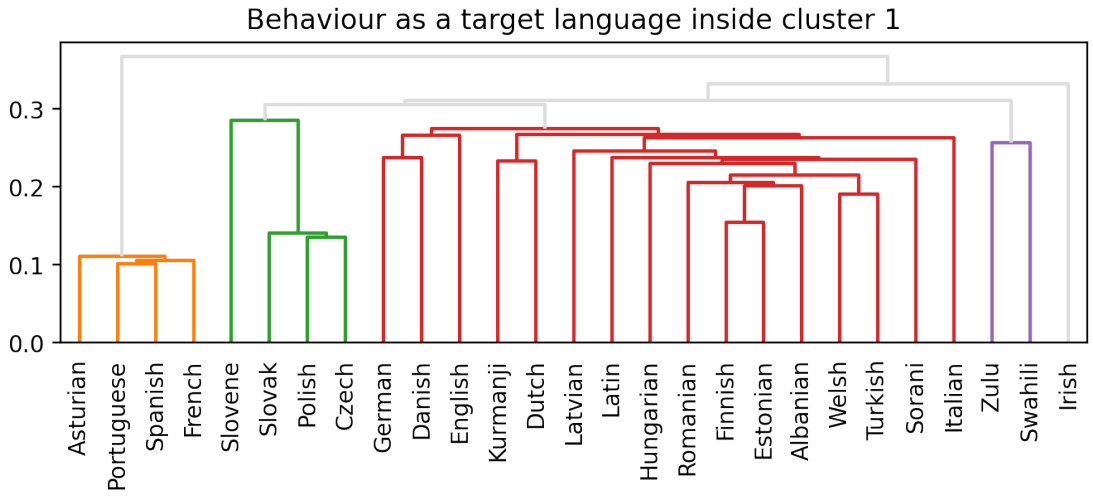

Figure 4: Dendrogram of the target languages, based on the transfer accuracy from all source languages as features for the target languages.

algorithm, correspond to closely related groups of languages. More precisely, the **orange** cluster contains Western Romance languages (Asturian, Portuguese, Spanish, and French), the **purple** cluster contains all the Bantu languages (Zulu and Swahili), and the **green** cluster contains Slavic languages: West Slavic languages (Slovak, Polish, and Czech) and slightly further the South Slavic language (Slovene). Finally, Irish is isolated and the **red** cluster contains all the remaining languages, even if distinct sub-clusters can be found: the (Finnish and Estonian) sub-cluster corresponds to Finnic languages and the (Romanian and Italian) sub-cluster contains the non-Western Romance languages. Other sub-clusters of the **red** cluster do not correspond to specific language families, like the (Kurmanji and Dutch) and the (Welsh and Dutch) sub-clusters.

From these results, we confirm that the morphological similarities of the languages are reflected in the model behavior during transfer. This indicates that our approach models morphological rules that can be transferred to related languages. However, it is clear that transfer in some clusters performs better than in others, though we are not yet able to provide explanations. Also, the performance is most likely influenced by the fact that the data in Sigmorphon2019 do not represent the full morphology of each language.

6. Conclusion

In this work, we use transfer to study the behavior of the ANNc analogy model when changing the axiomatic setting, the dataset, or the language of the analogies.

With results in 11 languages, we empirically confirmed that it is possible to model different axiomatic settings of analogy by changing the sets of permutations used when training ANNc. This highlights the importance of careful consideration on the axiomatic setting to use for data augmentation depending on the application, as it can significantly change model behavior. These results suggest that it is possible to determine the axiomatic setting matching a domain

from data. Indeed, if domain data containing valid and invalid analogies is available, an ANNc model can be learned and matched against multiple axiomatic settings to find the one fitting the domain. This kind of method could provide empirical arguments to define the notion of analogy in specific domains.

We also extended previous results on tranferability between languages and complemented it with transferability between datasets. Empirical results confirm previous hypotheses on the alphabet gap issue. We found that in many cases it is possible to use the proximity in the Wikipedia language families to predict the performance of transferred models, which confirm the transferability of morphological analogies between languages. These results suggest that analogies and transfer could be used to empirically study morphological similarities between languages. Such similarities can be useful in language learning, by selecting languages known by a learner and having similar morphology to a language to learn. They could also be used to automatically create a data-driven language classification.

Acknowledgments

Experiments presented in this paper were carried out using computational clusters equipped with GPU from the Grid'5000 testbed (see https://www.grid5000.fr). This research was partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation program under GA No 952215, and the Inria Project Lab "Hybrid Approaches for Interpretable AI" (HyAIAI).

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] M. Mitchell, Abstraction and analogy-making in artificial intelligence, Ann. N.Y. Acad. Sci. 1505 (2021) 79–101.

[3] H. Prade, G. Richard, Analogical proportions: Why they are useful in ai, in: 13th IJCAI, Survey Track, 2021, pp. 4568–4576.

[4] M. Bayoudh, H. Prade, G. Richard, Evaluation of analogical proportions through kolmogorov complexity, Knowledge-Based Systems 29 (2012) 20–30.

[5] D. Hofstadter, M. Mitchell, The copycat project: A model of mental fluidity and analogy-making, in: Fluid Concepts and Creative Analogies, 1995, pp. 205–267.

[6] Y. Lepage, Analogy and formal languages, in: 6th CFG and 7th CML, volume 53, 2001, pp. 180–191.

[7] Y. Lepage, De l'analogie rendant compte de la commutation en linguistique, Habilitation à diriger des recherches, Université Joseph-Fourier - Grenoble I, 2003.

[8] L. Miclet, S. Bayoudh, A. Delhay, Analogical dissimilarity: Definition, algorithms and two experiments in machine learning, JAIR 32 (2008) 793–824.

[9] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: 1st ICLR, Workshop Track, 2013.

[10] T. Mikolov, W.-T. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: NAACL, 2013, pp. 746–751.
[11] S. Alsaidi, A. Decker, P. Lay, E. Marquer, P.-A. Murena, M. Couceiro, A neural approach for detecting morphological analogies, in: IEEE 8th DSAA, 2021, pp. 1–10.
[12] S. Alsaidi, A. Decker, P. Lay, E. Marquer, P.-A. Murena, M. Couceiro, On the Transferability of Neural Models of Morphological Analogies, in: AIMLAI, ECML PKDD, volume 1524, 2021, pp. 76–89.
[13] C. Antic, Analogical proportions (2022).
[14] P.-A. Murena, M. Al-Ghossein, J.-L. Dessalles, A. Cornuéjols, Solving analogies on words based on minimal complexity transformation, in: 29th IJCAI, 2020, pp. 1848–1854.
[15] R. Fam, Y. Lepage, Tools for the production of analogical grids and a resource of n-gram analogical grids in 11 languages, in: 11th LREC, ELRA, 2018, pp. 1060–1066.
[16] R. Fam, Y. Lepage, Morphological predictability of unseen words using computational analogy., in: 24th ICCBR workshops, 2016, pp. 51–60.
[17] S. Lim, H. Prade, G. Richard, Solving word analogies: A machine learning perspective, in: 15th ECSQARU, volume 11726, 2019, pp. 238–250.
[18] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: EMNLP, 2014, pp. 1532–1543.
[19] P. Langlais, F. Yvon, P. Zweigenbaum, Improvements in analogical learning: Application to translating multi-terms of the medical domain, in: 12th EACL, ACL, 2009, pp. 487–495.
[20] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, R. Harshman, Using latent semantic analysis to improve access to textual information, in: SIGCHI, 1988, pp. 281–285.
[21] E. Marquer, S. Alsaidi, A. Decker, P.-A. Murena, M. Couceiro, A Deep Learning Approach to Solving Morphological Analogies, 2022. To appear in 30th ICCBR.
[22] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, 2019, ACL, 2019, pp. 4171–4186.
[23] A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations, in: NeurIPS, 2020, pp. 12449–12460.
[24] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: ICLR, OpenReview.net, 2021.
[25] R. Cotterell, C. Kirov, J. Sylak-Glassman, D. Yarowsky, J. Eisner, M. Hulden, The sigmorphon 2016 shared task–morphological reinflection, in: SIGMORPHON 2016, ACL, 2016, pp. 10–22.
[26] M. Karpinska, B. Li, A. Rogers, A. Drozd, Subcharacter information in japanese embeddings: when is it worth it?, in: RLSNA4NLP, ACL, 2018, pp. 28–37.
[27] E. Marquer, M. Couceiro, S. Alsaidi, A. Decker, Siganalogies - morphological analogies from Sigmorphon 2016 and 2019, 2022.
[28] A. D. McCarthy, E. Vylomova, S. Wu, C. Malaviya, L. Wolf-Sonkin, G. Nicolai, C. Kirov, M. Silfverberg, S. J. Mielke, J. Heinz, R. Cotterell, M. Hulden, The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection, in: 16th CRPPM workshops, ACL, 2019, pp. 229–244.

A. Coverage Between the Source and the Target Language

In Fig. 1, we can see the percentage of characters of a target language that are also present in the source language. In Fig. 2, we can see the dendrogram of the hierarchical clustering on the Jaccard index of the characters present in each pair of languages.

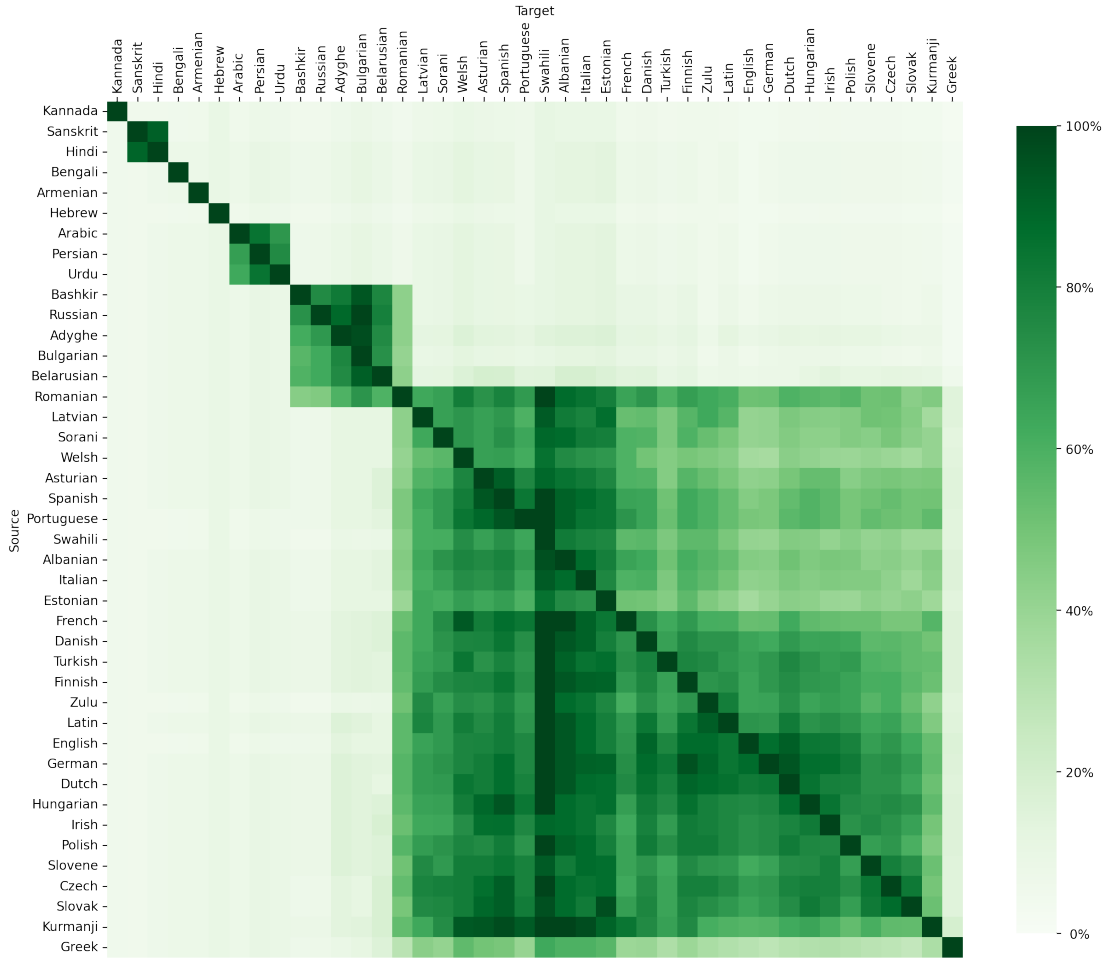

Figure 1: Coverage by the source language character vocabulary of the target language character vocabulary.

B. Transfer Performance Within the Largest Cluster of Languages

In Fig. 3, we can see the tree representing the language families of each of the languages in the lagest cluster (Albanian, Asturian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Hungarian, Irish, Italian, Kurmanji, Latin, Latvian, Polish, Portuguese, Romanian,

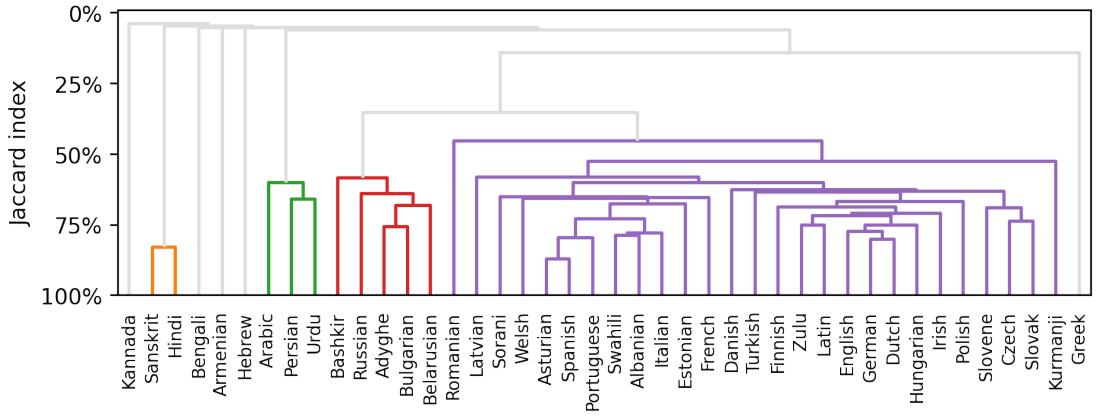

Figure 2: Dendrogram of the high resource languages in Sigmorphon2019 (except Basque and Uzbek), based on the Jaccard index between each pair of languages. With a threshold of 40% on the Jaccard and excluding singletons, four clusters (colored here) are found.

Slovak, Slovene, Sorani, Spanish, Swahili, Turkish, Welsh, and Zulu). The language families are extracted from the "Language family" field of the infobox in the Wikipedia page of each language.

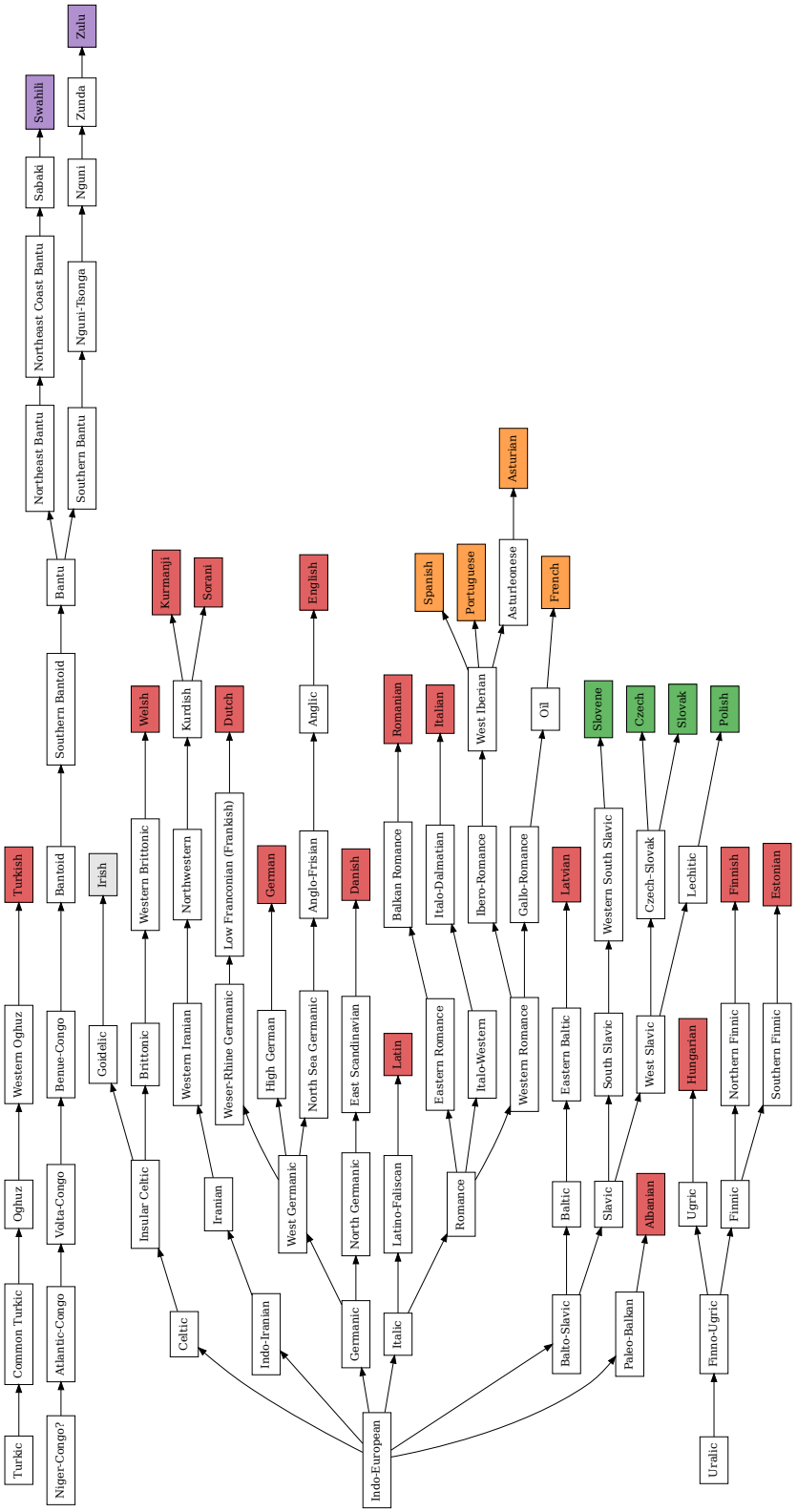

Figure 3: Trees of how the languages in cluster 1 relate based on their "Language family" according to Wikipedia. The Wikipedia page of each language contains an infobox (the area containing key information about the topic of the page), from which we extracted the "Language family" field. Languages are highlighted to match the clusters in the dendrogram of cluster 1.

Extraction of analogies between sentences on the level of syntax using parse trees

Yifei Zhou[1,*], Rashel Fam[1] and Yves Lepage[1]

[1]Waseda University, 2-7 Hibikino, Kitakyushu, 808-0135, Japan

Abstract

Example-based machine translation by analogy is an alternative approach to machine translation. Its principle is relatively simple, but the absolute number of analogies between sentences contained in the corpus is crucial for the overall quality of translation. The relative number of analogies is called the analogical density. The goal of this paper is to measure the analogical density of different aligned corpora. To this end, we extract analogies between sentences. Now, we use parse trees to represent sentences on the level of syntax. We report analogical densities for five different languages in an aligned multilingual corpus extracted from the Tatoeba resource, at the level of characters, words or parse trees.

Keywords

Sentence analogy, parse tree, example-based machine translation

1. Introduction

Analogy is known to be an essential skill in human cognition. It can be used to interpret or analyze words or sentences that are unfamiliar or have never been seen before [1, 2, 3, 4, 5]. In other words, analogy has the power to explain the unknown using the known. Analogy can play a role in natural language processing tasks such as machine translation [6, 7, 8], transliteration [9, 10] or question answering [11].

Example-Based Machine Translation (EBMT) by analogy implements a case-based reasoning approach to machine translation [12]. It generates translations relying on analogies in the source language and the target language after retrieval of similar sentences from a knowledge database. There, analogy exploits examples (cases) contained in the knowledge container (case base) to solve unknown cases.

By denoting $A : B :: C : D$ the analogical relationship between four sentences: A, B, C and D, Formula (1) defines sentence analogies in two languages with sentences which are translations of one another. $A : B :: C : D$ denotes a monolingual analogy in the source language and $A' : B' :: C' : D'$ is corresponding translation in the target language. Figure 1 instantiates Formula (1) on an example in English and French.

ICCBR Analogies'22: Workshop on Analogies: from Theory to Applications at ICCBR-2022, September, 2022, Nancy, France

*Corresponding author.

yifei.zhou@ruri.waseda.jp (Y. Zhou); fam.rashel@fuji.waseda.jp (R. Fam); yves.lepage@waseda.jp (Y. Lepage)

$$\begin{array}{ccccccc} A & : & B & :: & C & : & D \\ \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow \\ A' & : & B' & :: & C' & : & D' \end{array} \tag{1}$$

I like apples.	:	*I don't like apples.*	::	*I speak Swedish.*	:	*I don't speak Swedish.*
↕		↕		↕		↕
J'aime les pommes.	:	*Je n'aime pas les pommes.*	::	*Je parle le suédois.*	:	*Je ne parle pas le suédois.*

Figure 1: Two corresponding monolingual analogies between sentences in English and French

The number of analogies that exist in a given corpus is crucial for EBMT by analogy. Our objective in the present work is to estimate the number of analogies similar to the one shown in Figure 1, for various language pairs. Now, analogies can be extracted at various levels: surface form or syntax. To extract analogies automatically, we use vector representations of sentences based on the occurrence of characters, tokens, or branches in parse trees. We then count the number of extracted analogies and can compute the *analogical density* of the corpus. Although we do not conduct experiments in this paper, our intuition is that a higher number of analogies will lead to better translations in an EBMT system by analogy.

2. Related Work

2.1. Traditional Levels: Formal and Semantic Analogies

Formal analogies do not take into account the meaning or the syntax of sentences. Instead, the surface form, i.e., characters or words, are only taken into account. [13] uses *abc* : *abbccd* :: *efg* : *effggh* as an example to clarify what formal analogy is. The changes are only between characters and the strings bear no meaning. *walk* : *walked* :: *go* : *goed* is another instance of formal analogy: *goed* is not a valid English word form for the simple past tense form of *go*. However, on the level of form, the analogy holds: the suffix *-ed* has just been added at the end of the string *go*, as for *walk*.

In an analogy at the semantic level, the meaning attached to the strings is considered. For instance, *king* : *queen* :: *man* : *woman* is a classic example of semantic analogy [14]. It exhibits the male / female opposition. In contrast to the previous formal analogy, *walk* is to *walked* as *go* is to *went* is valid on the level of meaning, or rather grammar. Table 1 shows examples of analogies between sentences on one of the two levels of form or meaning, or both.

2.2. Between Form and Semantics: Syntax

In the present paper, we concentrate on analogies between sentences. In [15, 16, 17], a method to perform syntactic analysis of sentences, i.e., to obtain a syntax tree for a given sentence, has

Table 1
Example of analogies between sentences on the level of form and meaning

Analogy between sentences								Level	
								Form	Meaning
They work hard.	:	*He worked very hard.*	::	*They look happy.*	:	*He looked very happy.*		yes	yes
The boy speaks Thai.	:	*The girl goes to Thailand.*	::	*The actor spoke Chinese.*	:	*The actress went to China.*		no	yes
I talk to him.	:	*I talked to him.*	::	*I go to school.*	:	*I goed to school.*		yes	no

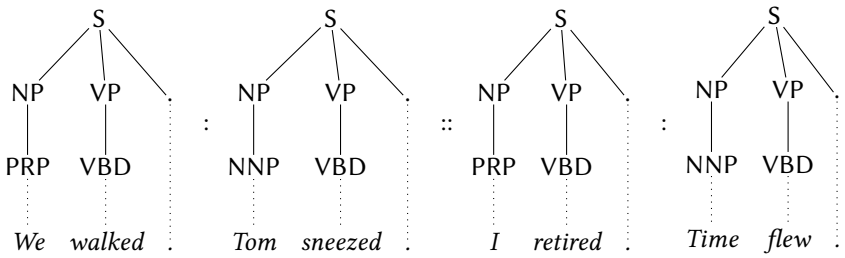

Figure 2: Analogy between sentences on the level of syntax using constituency representation

been described. It relied on the use of analogy. Similarly, an example of an analogy between syntactic trees is shown in [18, 19]. It corresponds to an active passive transformation between sentences: the analogy holds not only on the level of form, but also on the level of syntax.

[18, 19] show that syntactic representations of sentences can be used as yet another level to capture analogies between sentences, in addition to the formal and semantic levels. However, analogy on the level of syntax is different from both the formal and semantic levels. It is well known that grammaticality is independent from meaning, as illustrated by the classic example sentence: *Colorless green ideas sleep furiously* [20].

We propose to work on analogy at the level of syntax. Figure 2 is another example of a syntactic analogy between sentences. There, the sentences do not acceptedly create an analogy on the level of form or meaning, but they definitely make an analogy at the syntactic level: exchange of personal pronoun (PRP) with proper noun (NNP). Notice that, for the analogy to hold, the terminals (the words in the sentences) which should appear on the leaves in the parse trees are not considered.

3. Analogy on the Level of Syntax Using Parse Trees

While past studies concentrated on analogies on the formal level, the originality of this paper is to extract and count analogies between sentences on the level of syntax using parse trees. To this end, we develop two components. The first component computes vector representations. A sentence is represented by a feature vector counting the number of occurrences of all branches in any parse tree of a sentence from the corpus considered. The second component computes

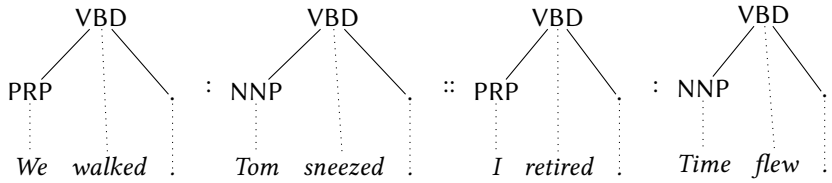

Figure 3: Analogy between sentences on the level of syntax using dependency representation

the ratio between these vectors of features on two given trees. The ratio between sentences is simply defined as the difference between their feature vectors.

3.1. Tree Representations

In computational linguistics, a parse tree is a tree that represents the syntactic structure of a sentence [21]. In constituency parse trees, the tree reflects the grouping of words in a sentence by constituents or phrases. In dependency parse trees, the branches show the dependency relationship between words. We use Universal Dependency parsers provided by the spaCy[1] library, for various languages, and converted all sentences in our corpora into dependency parse trees. The dependency parse trees of the sentences in Figure 2 are shown in Figure 3.

A sentence S can be represented by a feature vector $\overrightarrow{T_S}$ by counting the number of occurrences for all the branches found in its parse tree T_S. In Formula (2), the notation $|T_S|_{branch}$ stands for the number of times a *branch* appears in the parse tree T_S of sentence S.

$$\overrightarrow{T_A} = \begin{pmatrix} |T_A|_{VBD \to PRP} \\ |T_A|_{VBD \to NNP} \\ \vdots \\ |T_A|_{VBD \to .} \end{pmatrix} \tag{2}$$

3.2. Ratios between Trees

To extract analogies at the level of syntax, we calculate the ratio between trees. Formula (3) defines the ratio between sentences A and B as the difference between their vectors of syntactic features derived from their parse trees T_A and T_B.

$$A : B \triangleq \overrightarrow{T_A} - \overrightarrow{T_B} = \begin{pmatrix} |T_A|_{VBD \to PRP} - |T_B|_{VBD \to PRP} \\ |T_A|_{VBD \to NNP} - |T_B|_{VBD \to NNP} \\ \vdots \\ |T_A|_{VBD \to .} - |T_B|_{VBD \to .} \end{pmatrix} \tag{3}$$

3.3. Conformity of Ratios between Trees

An analogy $A : B :: C : D$ is satisfied by checking the equality of ratios. Formula (4) defines it. For the computation of ratios between vectors and for checking for equality of ratios, we rely

[1] spaCy: https://spacy.io/

on the Python library `Nlg`[2] [22]. In this way, we extract all analogies between all parse trees corresponding to all sentences contained in our corpus.

$$A : B :: C : D \quad \overset{\Delta}{\iff} \quad \overrightarrow{T_A} - \overrightarrow{T_B} = \overrightarrow{T_C} - \overrightarrow{T_D} \tag{4}$$

3.4. Analogical Clusters

An analogical cluster is defined as a set of pairs of sentences with exactly the same ratio [23] (see definition in Formula (5)). The Python library `Nlg` can be used to extract all analogical clusters from a set of objects represented by feature vectors. We apply it for the extraction of analogical clusters between sentences, at the level of syntax. The larger an analogical cluster, the more regular the transformations between the sentences in the clusters.

$$\begin{array}{l} A_1 : B_1 \\ A_2 : B_2 \\ \quad \vdots \\ A_n : B_n \end{array} \quad \overset{\Delta}{\iff} \quad \forall (i,j) \in \{1, \ldots, n\}^2, \;\; A_i : B_i :: A_j : B_j \tag{5}$$

4. Experiments and Results

4.1. Data Used

We use the Tatoeba[3] corpus. It is a collection of sentences in more than 100 languages. Here, we use five language parts from the Tatoeba corpus: English, French, German, Polish and Finnish. The sentences we used are aligned across all five languages, they are parallel sentences that correspond to each other. Table 2 gives some statistics on this corpus. For each language, we have around eight thousand sentences. English has the lowest number of types and Finnish has the largest one among the five languages. Hapaxes are words that appear only once in a corpus. Here, we observe that English has the smallest number of hapaxes with less than 60% while Finnish has the highest percentage with over than 70%. We verify again that languages with higher morphological richness tend to have a higher number of types and hapaxes. In interest to us is the conjecture that we should extract more analogies from a language with a higher Type-Token Ratio (TTR) since type-token ratio measures lexical richness.

4.2. Metrics

To evaluate the number of analogies between sentences contained in a corpus, two metrics used in [24] are considered.

4.2.1. Analogical Density

Formula (6) defines *analogical density* as the ratio of the number of actual analogies N_{nlg} and N_s^4. If the total number of sentences in the corpus is N_s, N_s^4 is the number of possibilities of

[2] `Nlg`: http://lepage-lab.ips.waseda.ac.jp/en/projects/kakenhi-15k00317/
[3] Tatoeba: https://tatoeba.org/

Table 2
Statistics on Tatoeba corpus

Language	Number of			Average length of		TTR	Hapaxes (%)
	lines	tokens	types	token	type		
en	7,964	40,493	6,839	4.23±2.21	6.48±2.24	0.17	58.47
fr	7,964	43,563	8,581	4.62±2.71	7.46±2.58	0.20	64.10
de	7,964	41,017	8,673	4.96±2.56	7.55±2.93	0.21	63.07
pl	7,964	32,816	10,956	5.44±2.81	7.47±2.48	0.33	70.70
fi	7,964	31,152	11,270	6.09±2.95	8.10±2.90	0.36	72.17

filling in the analogy pattern with any four sentences (with possible repetition) from the corpus. As there are 8 equivalent forms of analogies [25], this should be divided by 8 to consider only individual analogies. Because the denominator is a power of 4, values for density are usually numbers of the order of 10^{-9} or 10^{-12}.

$$D_{\text{nlg}} = \frac{N_{\text{nlg}}}{\frac{1}{8} \times N_s^4} \tag{6}$$

4.2.2. Proportion of Sentences Appearing in Analogies

Formula (7) calculates the proportion of sentences appearing in analogies by dividing the number of sentences appearing in at least one analogy (N_{s_nlg}) by the total number of sentences in the corpus (N_s). This makes a percentage.

$$P = \frac{N_{\text{s\_nlg}}}{N_s} \tag{7}$$

4.3. Results and Analysis

We carry out experiments on the extraction of analogies between sentences both on the level of surface form and syntax. On the level of form, each sentence is tokenised using two different tokenisation schemes: character or word. On the level of syntax, we extract sentence analogies from a corpus by using parse trees, as described in Section 3. We also concatenate formal feature vectors with syntax feature vectors to combine the two levels. We do not conduct experiments using `char` and `word` features at the same time, because they both work on the formal level.

4.3.1. Number of Analogical Clusters

Table 3 gives the number of analogical clusters extracted from our five languages based on different feature vectors. We observe that the number of analogical clusters extracted based on syntactic trees is hundreds or thousands times larger than on the level of characters or words. Analogical clusters extracted by the combination of `char` and `tree` or `word` and `tree` are of course smaller than if only one feature is considered. When using only the `tree` feature, Finnish has a significantly higher number of analogical clusters in comparison to the other

Table 3
Number of extracted analogical clusters from different features: characters (`char`), words (`word`) and syntax (`tree`)

Language	Feature used					
	`char`		✓	✓		
	`word`				✓	✓
	`tree`	✓		✓		✓
en		502,182	774	333	325	251
fr		1,712,538	546	164	290	131
de		939,892	822	424	384	288
pl		2,246,054	860	381	333	205
fi		5,510,699	692	355	332	242

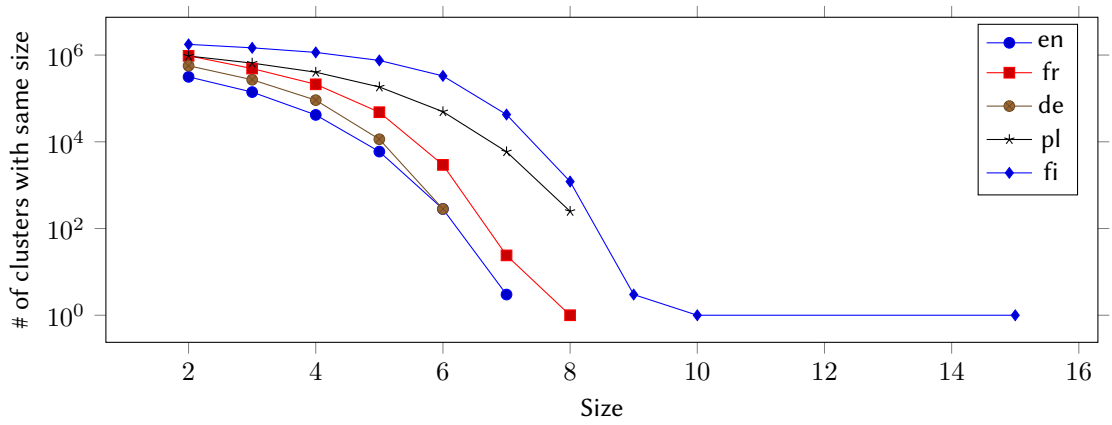

Figure 4: Number of extracted analogical clusters with the same size on the level of syntax among different languages. Caution: log scale on the y axis

languages, followed by Polish and French. Except for `char`, we observe that German always has the highest number of analogical clusters (except for `char` where it is second).

In addition, we draw the distribution of the number of analogical clusters with the same size extracted from syntactic features for our five languages in Figure 4. Although the numbers of extracted analogical clusters with the same size vary across languages, the overall trend is consistent.

4.3.2. Number of Analogies

The analogical density of the corpus is presented in Table 4. It indicates how many analogies can be extracted in the five different languages and how many sentences can be covered by these extracted analogies.

We observe that the number of analogies extracted on the level of syntax is thousands times more than on the level of form. Basically, on the level of syntax, Finnish has the highest

Table 4
Analogical densities for five different languages. The number of sentences is the number of sentences appearing in the extracted analogies. Notice the difference in orders of magnitude from `char` and `word` (10^{-12}) to `tree` (10^{-9}).

Feature	Language	Number of		Density		
		analogies	sentences	D_{nlg}		P (%)
`char`	en	985	723	1.96		9.08
	fr	679	486	1.35		6.10
	de	1,102	573	2.19	$\times 10^{-12}$	7.19
	pl	1,164	665	2.31		8.35
	fi	906	506	1.80		6.35
`word`	en	452	372	0.90		4.67
	fr	442	288	0.88		3.62
	de	603	328	1.20	$\times 10^{-12}$	4.12
	pl	559	281	1.11		3.53
	fi	580	248	1.15		3.11
`tree`	en	918,412	5,337	1.83		67.01
	fr	3,605,667	5,523	7.17		69.35
	de	1,786,002	5,336	3.56	$\times 10^{-9}$	67.00
	pl	6,369,718	6,343	12.68		79.65
	fi	20,027,663	6,999	39.83		87.88

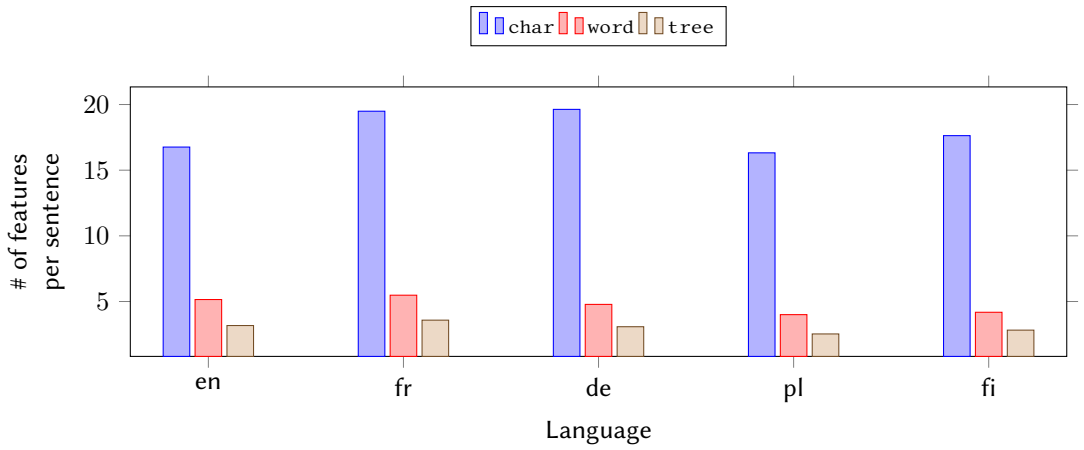

Figure 5: Number of features per sentence in extracted analogies on different levels

analogical density and the sentences that appear in analogy account for 88 percent of the whole corpus.

Figure 5 shows the number of features per sentence appearing in the extracted analogies on different levels. It is obvious that compared to the formal level, we extract more analogies on the syntactic level, given the smaller vector representations using parse trees.

Table 5
Example results of analogies in different languages extracted by combining formal and syntactic features

Lang.	Example results of sentence analogies			
en	*I bought a new car.*	: *He bought a new car.*	:: *I left the door open.*	: *He left the door open.*
fr	*Il regarde la télévision.*	: *Je regarde la télévision.*	:: *Il joue au tennis tous les jours.*	: *Je joue au tennis tous les jours.*
de	*Sie hat einen Hund.*	: *Sie hat einen Brief geschrieben.*	:: *Ich habe einen Hund.*	: *Ich habe einen Brief geschrieben.*
pl	*Jestem bohaterem.*	: *Jestem nauczycielem.*	:: *Nie jestem bohaterem.*	: *Nie jestem nauczycielem.*
fi	*Onko sinulla autoa?*	: *Onko sinulla veljiä tai siskoja?*	:: *Minulla ei ole autoa.*	: *Minulla ei ole veljiä tai siskoja.*

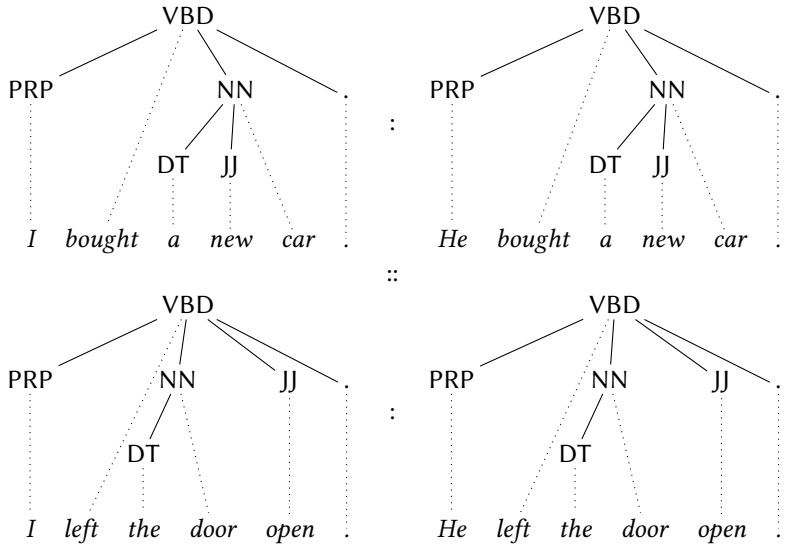

Figure 6: Parse trees of the English sentence analogy given in Table 5

4.4. Example Results of Analogies in Different Languages

In Table 5, we list some example results of sentence analogies that we extracted from the corpus in the combination of formal and syntactic features. Figure 6 plots the syntactic structure behind the first English example in Table 5.

5. Further Discussion

5.1. Analogical Grids

An analogical grid is a matrix where any four terms picked out from any two rows and any two columns is an analogy [26]. Formula (8) gives the definition of an analogical grid. The size of

We won.	:	*Tom won.*	:	*You won.*
We survived.	:	*Tom survived.*	:	*You survived.*
	:	*Tom drank too much.*	:	*You drank too much.*
We volunteered.	:	*Tom volunteered.*	:	

Figure 7: Example of an analogical grid in English extracted by combining the `tree` and `word` features

Table 6
Analogical grids extracted from different feature vectors in varying languages

Feature	Language	# of grids	Avg. size	Avg. saturation (%)
`tree` ∩ `char`	en	112	4.81	99.8
	fr	50	5.34	99.0
	de	82	6.91	98.9
	pl	94	5.72	99.4
	fi	78	5.42	99.5
`tree` ∩ `word`	en	82	4.87	99.7
	fr	39	5.51	98.7
	de	63	6.52	98.9
	pl	55	5.87	99.1
	fi	48	5.38	99.7

an analogical grid is defined as the product of its number of rows by its number of columns. As shown in Figure 7, an analogical grid may have empty cells. Thus, we can also characterise an analogical grid by the number of non-empty cells in it. This is its *saturation.* It is the ratio between the number of non-empty cells and the size of the grid.

$$\begin{array}{ccc} G_1^1 : G_1^2 : \cdots : G_1^m \\ G_2^1 : G_2^2 : \cdots : G_2^m \\ \vdots \quad \vdots \qquad\quad \vdots \\ G_n^1 : G_n^2 : \cdots : G_n^m \end{array} \quad \overset{\Delta}{\Longleftrightarrow} \quad \begin{array}{c} \forall (i,k) \in \{1,\ldots,n\}^2, \\ \forall (j,l) \in \{1,\ldots,m\}^2, \\ G_i^j : G_i^l :: G_k^j : G_k^l \end{array} \tag{8}$$

We have extracted analogical grids on the level of syntax combined with character features or word features. By extracting the analogical grids, we hope to get a more compact view of how sentences are related to each other by analogies. Based on Table 6, we observe that English has the highest number of analogical grids but also the smallest average size of analogical grids. German has the largest average size of analogical grids. The average saturation of the extracted analogical grids is all around 99 % which means the analogical grids extracted from our corpus are very dense.

5.2. Extracted analogies for different language pairs

For any language pair from the five languages in the aligned corpora, we can extract bilingual analogies by taking monolingual analogies where sentences correspond by translation. This kind of data, i.e., bilingual analogies, can then be exploited in an EBMT system by analogy.

Table 7
Number of extracted bilingual analogies for different language pairs

Feature	en-fr	en-de	en-pl	en-fi	fr-de	fr-pl	fr-fi	de-pl	de-fi	pl-fi
`char`	64	77	50	34	88	54	40	129	42	38
`tree` ∩ `char`	28	30	14	18	46	22	14	43	16	8
`word`	48	24	20	24	48	44	30	86	30	22
`tree` ∩ `word`	24	20	12	12	30	20	10	26	12	6

Table 7 counts the number of extracted bilingual analogies for different language pairs on the formal and syntactic levels. Because these are intersections, the number of bilingual analogies is of course smaller than the number of independent monolingual analogies for any of the languages in the language pair.

6. Conclusion

We proposed to extract analogies between sentences based on their syntactic structure. Experiments were carried out using Universal Dependency parse trees that allow us to compare across five different European languages. The parse trees were converted into feature vectors, the features of which were the types of branches, from which we removed the lexical information. We measured the analogical density at the syntactic level and crossed with the results at the character or word levels.

We found that the number of analogies extracted on the syntactic level is hundreds or thousands times larger than the one on the formal level, which leads to a thousand times higher analogical density. We already started extracting analogical grids to have a more compact view of how sentences are related to each other.

In this paper, we used the number of occurrences of branches in dependency representations as features to get a vector representation of sentences. Similar work could be carried out with constituency representations, if constituency parsers comparable across languages would be available. The ultimate goal of the work presented here, is to not only to extract monolingual analogies, but bilingual analogies between sentences, because they can be used by an EBMT system by analogy.

References

[1] H. Paul, Prinzipien der Sprachgeschichte, Niemayer, Tübingen, 1920.
[2] F. de Saussure, Cours de linguistique générale, Payot, Paris, 1916.
[3] L. Bloomfield, Language, Henry Holt, 1933.
[4] A. Welcomme, Hermann Paul et le concept d'analogie, CÍRCULO de Lingüística Aplicada a la Comunicación (clac) 43 (2010) 49–122.
[5] R. Fam, A. Purwarianti, Y. Lepage, Plausibility of word forms generated from analogical grids in Indonesian, in: Proceedings of the 16th International Conference on Computer Applications (ICCA–2018), Yangon, Myanmar, 2018, pp. 179–184.

[6] M. Nagao, A framework of a mechanical translation between Japanese and English by analogy principle., in: Artificial and human intelligence, 1984, p. 351–354.

[7] Y. Lepage, E. Denoual, The 'purest' EBMT system ever built: No variables, no templates, no training, examples, just examples, only examples, in: Workshop on example-based machine translation, Phuket, Thailand, 2005, pp. 81–90. URL: https://aclanthology.org/2005.mtsummit-ebmt.11.

[8] S. Dandapat, S. Morrissey, S. K. Naskar, H. Somers, Mitigating problems in analogy-based EBMT with SMT and vice versa: A case study with named entity transliteration, in: Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation, Institute of Digital Enhancement of Cognitive Processing, Waseda University, Tohoku University, Sendai, Japan, 2010, pp. 365–372. URL: https://aclanthology.org/Y10-1041.

[9] P. Langlais, Mapping source to target strings without alignment by analogical learning: A case study with transliteration, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 684–689. URL: https://aclanthology.org/P13-2120.

[10] R. Rhouma, P. Langlais, Fourteen light tasks for comparing analogical and phrase-based machine translation, in: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 444–454. URL: https://aclanthology.org/C14-1043.

[11] A. Diallo, M. Zopf, J. Fürnkranz, Learning analogy-preserving sentence embeddings for answer selection, CoRR abs/1910.05315 (2019). URL: http://arxiv.org/abs/1910.05315. arXiv:1910.05315.

[12] B. Collins, H. Somers, Recent Advances in Example-Based Machine Translation, Springer Netherlands, Dordrecht, 2003, pp. 115–153. URL: https://doi.org/10.1007/978-94-010-0181-6_4. doi:10.1007/978-94-010-0181-6_4.

[13] D. R. Hofstadter, Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought, Basic Books, Inc., USA, 1996.

[14] T. Mikolov, W.-T. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013), Association for Computational Linguistics, Atlanta, Georgia, 2013, pp. 746–751. URL: http://www.aclweb.org/anthology/N13-1090.

[15] Y. Lepage, S.-i. Ando, Saussurian analogy: a theoretical account and its application, in: COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics, 1996, pp. 717–722. URL: https://aclanthology.org/C96-2121.

[16] S.-I. Ando, Y. Lepage, Linguistic structure analysis by analogy: Its efficiency, in: Proceedings of NLPRS-97, Phuket, 1997, pp. 401–406. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.8231. doi:10.1.1.50.8231.

[17] Y. Lepage, S.-I. Ando, S. Akamine, H. Iida, An annotated corpus in Japanese using Tesnière's structural syntax, in: Processing of Dependency-Based Grammars, 1998, pp. 109–115. URL: https://aclanthology.org/W98-0513.

[18] N. Stroppa, F. Yvon, An analogical learner for morphological analysis, in: Proceedings

of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005), Association for Computational Linguistics, Ann Arbor, Michigan, 2005, pp. 120–127. URL: https://aclanthology.org/W05-0616.

[19] N. Stroppa, F. Yvon, Analogical learning and formal proportions: Definitions and methodological issues, ENST Paris report (2005).

[20] N. Chomsky, The Logical Structure of Linguistic Theory, Springer US, 1975. URL: https://books.google.co.jp/books?id=1D66ktXOITAC.

[21] I. Chiswell, W. Hodges, Mathematical Logic, Oxford University Press, Inc., USA, 2007.

[22] R. Fam, Y. Lepage, Tools for the production of analogical grids and a resource of n-gram analogical grids in 11 languages, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018, pp. 1060–1066. URL: https://aclanthology.org/L18-1171.

[23] Y. Lepage, C. L. Goh, Towards automatic acquisition of linguistic features, in: Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009), Northern European Association for Language Technology (NEALT), Odense, Denmark, 2009, pp. 118–125. URL: https://aclanthology.org/W09-4618.

[24] R. Fam, Y. Lepage, A study of analogical density in various corpora at various granularity, Information 12 (2021) 314. URL: https://doi.org/10.3390/info12080314. doi:10.3390/info12080314.

[25] Y. Lepage, Languages of analogical strings, in: Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000), volume 1, Saarbrücken, 2000, pp. 488–494. URL: https://aclanthology.org/C00-1071.

[26] R. Fam, Y. Lepage, Morphological predictability of unseen words using computational analogy, CEUR Workshop Proceedings 1815 (2016) 51–60.

CoAT-APC: When Analogical Proportion-based Classification Meets Case-based Prediction

Fadi Badra[1,*], Marie-Jeanne Lesot[2,*]

[1]Université Sorbonne Paris Nord, LIMICS, Sorbonne Université, INSERM, Bobigny, France
[2]Sorbonne Université, CNRS, LIP6, Paris, France

Abstract

This paper proposes to view analogical proportion-based classification as a special type of case-based prediction algorithm, in which (i) cases are *differences* between two instances, and (ii) only maximally similar cases are compared. It then proposes to tweak the CoAT case-based prediction algorithm in order to implement these two key design principles. The resulting analogical proportion-based classifier CoAT-APC shows a performance comparable to state-of-the-art analogical proportion-based classifiers, while implementing a different transfer strategy, based on the minimization of a dataset complexity measure, as opposed to a rule-based approach. Experimental results show the usefulness of combining these two design principles and suggest that the rule-based transfer strategy of analogical proportion-based classifiers has comparatively little impact on the performance of the system.

Keywords

analogical proportion-based classification, case-based prediction, interactions between case-based and analogical reasoning

1. Introduction

Case-based prediction [1, 2] consists in predicting the outcome (the label, in classification tasks) of a new case directly from its comparison with a set of cases retrieved from a case base and some similarity measures, without any attempt to learn a model of the observed data prior to the inference. In case-based prediction methods, cases are assumed to be pairs (situation, outcome) and the predicted outcome is the one that best enforces a compatibility requirement, according to which outcome similarities in the resulting case base should be compatible with the corresponding situation similarities. This principle can be expressed in numerous ways, leading to a large variety of case-based prediction algorithms, as discussed later in the paper. They mainly differ in their transfer strategy, that can for instance rely on rule-based or optimization-based approaches.

On the other hand, analogical proportion-based classification (abbreviated APC in the rest of the paper, see e.g. [3, 4, 5, 6, 7]) proposes to exploit the principle of analogical reasoning, based on statements of the form "**a** is to **b** as **c** is to **d**", to predict the label of new instances: when

ICCBR Analogies'22: Workshop on Analogies: from Theory to Applications at ICCBR-2022, September, 2022, Nancy, France

*Corresponding author.
badra@univ-paris13.fr (F. Badra); marie-jeanne.lesot@lip6.fr (M. Lesot)
0000-0002-2437-8230 (F. Badra); 0000-0002-3604-6647 (M. Lesot)

an analogical proportion holds on the instance descriptions, it is inferred that an analogical proportion also holds on their associated labels.

This paper proposes to compare these two classification paradigms and to study APC from the point of view of case-based prediction. More precisely, it shows that APC can be interpreted as a special kind of case-based prediction method, that applies a rule-based transfer strategy and implements the two following design principles:

[P1] Cases are *differences* between two instances of the considered instance set.
[P2] Only maximally similar cases are compared.

The second principle can be relaxed to comparing only the most similar cases, beyond the maximally similar ones, as discussed later in the paper.

The paper then investigates why analogical proportion-based classifiers exhibit better performance compared to other case-based prediction methods. To do so, a new case-based prediction method is proposed, called CoAT-APC, that tweaks the CoAT case-based prediction algorithm [8, 9] to implement the two principles [P1] and [P2] with the relaxed version of the latter. The resulting algorithm is an analogical proportion-based classifier, in which the rule-based transfer strategy is replaced by CoAT's optimization-based transfer strategy.

Experiments conducted on a variety of benchmark data sets, both of Boolean and numerical types, show that CoAT-APC offers performances comparable to state-of-the-art analogical proportion-based classifiers, and that complying with the two principles [P1] and [P2] significantly improves the performance of the original CoAT algorithm. This suggests that the two principles [P1] and [P2] do play a major role in their success, whereas their rule-based transfer strategy has little impact on their efficiency, and can be replaced without harm with another transfer strategy.

The paper is structured as follows: Section 2 recalls the main definitions about analogical proportion-based classifiers. Section 3 considers case-based prediction methods and in particular reviews their main prediction strategies. Section 4 shows that APC can be formulated as a case-based prediction method with rule-based transfer strategy. Section 5 presents the proposed exploitation of this view in the CoAT-APC algorithm, that modifies the CoAT case-based prediction method with optimization transfer strategy [8, 9] in order to implement the two key design principles of APC. Section 6 presents some experiments to validate the approach. Section 7 concludes the paper and discusses some directions for future work.

2. Analogical Proportion-based Classification

Analogical proportion-based classifiers have shown competitive results in classification and recommendation tasks, see e.g. [10, 3, 11, 12, 5]. They apply the principle of analogical reasoning [13], based on statements of the form "**a** is to **b** as **c** is to **d**", called analogical proportion, and written $\mathbf{a} : \mathbf{b} :: \mathbf{c} : \mathbf{d}$. More precisely, the analogical inference is applied in a classification setting to state that if an analogical proportion holds on the instance descriptions, then an analogical proportion can be inferred on their associated class labels: formally, denoting f the underlying, unknown, labelling function, one can derive from $\mathbf{a} : \mathbf{b} :: \mathbf{c} : \mathbf{d}$ that $f(\mathbf{a}) : f(\mathbf{b}) :: f(\mathbf{c}) : f(\mathbf{d})$. Let D be a data set containing a set of instances **a**,**b**,**c**, ... with

their associated labels $f(\mathbf{a}), f(\mathbf{b}), f(\mathbf{c}), \ldots$ To predict the value $f(x)$ for a new instance x, an analogical proportion-based classifier considers all triples (**a**,**b**,**c**)$\in D^3$ for which $\mathbf{a}:\mathbf{b}::\mathbf{c}:x$ holds, and the equation $f(\mathbf{a}):f(\mathbf{b})::f(\mathbf{c}):y$ has a solution. This set of triples is called the *analogical root* of x [4]. The predicted label for the new instance x is then the result of a majority vote among the potential solutions y. Yet it can be the case that the analogical root is empty: the previous classifier can then be extended to consider approximate analogy, relying on the notion of analogical dissimilarity [4]. The latter is defined as a function $AD(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d})$ that quantifies the extent to which the quadruplet is far from satisfying an analogical proportion: AD is such that $AD(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}) = 0$ iff $\mathbf{a}:\mathbf{b}::\mathbf{c}:\mathbf{d}$ and satisfies constraints on argument permutation and a triangular inequality [10]. For real or Boolean values, it can for instance be defined as the sum of the componentwise $AD(a,b,c,d) = \|(a-b)-(c-d)\|_1$. If the analogical root of x is empty, the search for potential solutions is extended to triples $(\mathbf{a},\mathbf{b},\mathbf{c})$ with the k least values of $AD(\mathbf{a},\mathbf{b},\mathbf{c},x)$ and for which the equation $f(\mathbf{a}):f(\mathbf{b})::f(\mathbf{c}):y$ has a solution. The predicted label is the result of a majority vote among the potential solutions y.

3. Case-Based Prediction: A Comparative Study

This section recaps the general principles of case-based prediction methods and reviews the main transfer strategies they rely on, proposing to distinguish between four categories discussed in turn in Sections 3.2 to 3.5.

3.1. Definition and Notations

Case-based prediction typically considers the following setting: $\mathcal{S}$ denotes an input space and $\mathcal{O}$ an output space. An element of $\mathcal{S}$ is called a *situation*, and an element of $\mathcal{O}$ is called an *outcome*, or a result. A finite set $CB = \{(s_1, r_1), \ldots, (s_n, r_n)\}$ of elements in $\mathcal{S} \times \mathcal{O}$ is called a *case base*. For legibility, and abusing the notation, cases and outcomes are sometimes denoted with their corresponding situation as subscript: an element $c_s = (s, r_s) \in CB$ is called a *source case*. In addition, σ_S and σ_R respectively denote similarity measures on situations and on outcomes. For a new case $c_t = (t, r_t)$ whose outcome r_t is to be predicted, a common decomposition of the case-based inference involves three main tasks [14]:

- *Retrieval*: retrieve from CB a set of source cases $c_s = (s, r_s)$;
- *Mapping*: for each retrieved situation s, compute the similarity $\sigma_S(s,t)$ between s and the target situation t;
- *Transfer*: estimate the similarities $\sigma_R(r_s, r_t)$ on outcomes from the similarities $\sigma_S(s,t)$ on situations.

In the transfer task, as illustrated in the diagram of Fig. 1, a plausible inference is triggered in order to estimate the similarity $\sigma_R(r_s, r_t)$ on outcomes from the similarity $\sigma_S(s,t)$ on situations: it applies the principle according to which if two situations are similar, then it is plausible that their outcomes are also similar.

For a new situation t, case-based prediction is then a search, among all potential outcomes $r \in \mathcal{O}$, for the outcome r_t that makes the plausible inference most likely to succeed when the

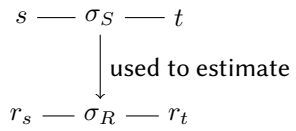

Figure 1: Schematic view of the transfer task in case-based prediction: the similarity relation on situations used to estimate the similarity relation on outcomes.

new case is added to the source case to build the augmented case base $CB \cup \{(t, r_t)\}$. To find this outcome, case-based prediction methods express the plausible inference as a compatibility requirement on the resulting similarity relations: when the new case is compared to the retrieved cases, the outcome similarities should be compatible with the observed situation similarities. In the following, $\hat{c}_t = (t, r)$ denotes a potential new case formed by choosing the outcome $r \in \mathcal{O}$ for the new case.

Different prediction strategies can be found in the literature to express this compatibility requirement between the two similarity measures. The next sections propose to distinguish between four categories of transfer strategies, respectively named transfer by rule-based voting, by constraint, by evidence support and by optimization.

3.2. Transfer by Rule-based Voting

A first type of transfer strategy relies on rules providing information on relations between the similarity measures σ_S and σ_R, expressing that when σ_S takes value α, the resulting similarity level for σ_R is β: these rules can be written $(\sigma_S = \alpha) \rightarrow (\sigma_R = \beta)$ and can be expressed in various forms, such as adaptation rules [15, 16, 17, 18], dependencies between problem and solution features [19], co-variations [20] or fuzzy rules [21] to name a few. The prediction strategy consists in triggering the rules on pairs of cases involving the new case using a kind of similarity-based inference, as detailed below, in order to derive potential outcomes for the new case.

The proposed outcome r_t is obtained by a majority vote on the set of outcomes r derived from the rules. Triggering a rule consists in performing a similarity-based inference (SBI), applying variants of the modus ponens schema [21, 22, 23]: for a retrieved case $c_s = (s, r_s)$ and a potential new case $\hat{c}_t = (t, r)$, triggering the rule $(\sigma_S = \alpha) \rightarrow (\sigma_R = \beta)$ on the pair of cases $(c_s, \hat{c}_t)$ is of the form

$$\frac{(\sigma_S = \alpha) \rightarrow (\sigma_R = \beta) \quad \sigma_S(s,t) \approx \alpha}{\sigma_R(r_s, r) \approx \beta} \tag{SBI}$$

It can be noted that it is often the case that the similarity measures σ_S and σ_R are unknown, or difficult to assess globally on the training data. One strategy then consists in working with some local approximations $\widetilde{\sigma_S}$ of σ_S and $\widetilde{\sigma_R}$ of σ_R that are known to be compatible for some pairs of cases of the case base. The resulting rules $(\widetilde{\sigma_S} = \alpha) \rightarrow (\widetilde{\sigma_R} = \beta)$ are *adaptation rules*, that may be acquired from an expert [24], from the user [25] or learned from data [26, 27, 15, 16, 18].

3.3. Transfer by Continuity Constraints

Another strategy consists in expressing the compatibility requirement between the two similarity measures σ_S and σ_R as a set of continuity constraints à la Lipschitz [28], for instance of the form $\sigma_R(r_s, r_t) \geq h(\sigma_S(s,t))$, where h is a transformation function that contains the provided information about the relation between σ_S and σ_R. Examples include similarity profiles [29], or gradual rules or certainty rules [30, 31, 32]. Such constraints are used to reduce the set of potential outcomes, excluding the ones that violate them. The predicted outcome is chosen among the potential outcomes that are consistent with all constraints.

3.4. Transfer by Evidence Support

A more data-driven type of approach consists in using a joint similarity measure to estimate for each pair of cases $(c_s, \hat{c_t})$ how compatible the similarity relation $\sigma_R(r_s, r)$ is with the similarity relation $\sigma_S(s,t)$. Examples include the k-Nearest Neighbor algorithm or the possibilistic instance-based learning approach [28, 33, 34]. In these approaches, a new case is considered possible if the existence of a similar case is confirmed by observation. The value of the joint similarity measure is interpreted as a degree of *confirmation*, or *evidence support* that the new case is supported by the retrieved source cases. The predicted outcome r_t is the one for which the maximal compatibility would be observed with a source case.

3.5. Transfer by Optimization

In most case-based prediction approaches, the compatibility of σ_R with σ_S is evaluated on the pair of cases $(c_s, \hat{c_t})$ for each retrieved case c_s, and the results are combined in order to find the most plausible outcome r for the new case. A recent work [8] proposes to define a global indicator that measures the compatibility of σ_R with σ_S on the whole case base: the prediction then consists in minimizing the value of a dataset complexity indicator when augmented with the new case and its candidate associated outcome. This principle is implemented in the CoAT, for **Co**mplexity-based **A**nalogical **T**ransfer, algorithm [8, 9]. In the CoAT method, the compatibility of σ_R with σ_S is measured from an ordinal point of view on the whole case base CB, by checking if σ_R orders the cases in the same manner as σ_S. The following continuity constraint is tested on each triple of cases (c_0, c_i, c_j), with $c_0 = (s_0, r_0)$, $c_i = (s_i, r_i)$, and $c_j = (s_j, r_j)$:

$$\text{if } \sigma_S(s_0, s_i) \geq \sigma_S(s_0, s_j), \text{ then } \sigma_R(r_0, r_i) \geq \sigma_R(r_0, r_j) \qquad (C)$$

The constraint (C) expresses that anytime a situation s_i is more similar to a situation s_0 than situation s_j, this order should be preserved on outcomes. A triple (c_0, c_i, c_j) does *not* satisfy the constraint if situation s_i is more similar to s_0 than situation s_j for situations, but less similar for outcomes, *i.e.*, when $\sigma_S(s_0, s_i) \geq \sigma_S(s_0, s_j)$ and $\sigma_R(r_0, r_i) < \sigma_R(r_0, r_j)$. Such a violation of the constraint is called an *inversion of similarity*. A global indicator $\Gamma(\sigma_S, \sigma_R, CB)$ is introduced, that counts the total number of inversions of similarity observed on a case base CB:

$$\Gamma(\sigma_S, \sigma_R, CB) = |\{((s_0, r_0), (s_i, r_i), (s_j, r_j)) \in CB \times CB \times CB \text{ such that } \sigma_S(s_0, s_i) \geq \sigma_S(s_0, s_j) \text{ and } \sigma_R(r_0, r_i) < \sigma_R(r_0, r_j)\}|$$

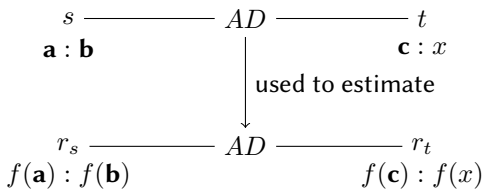

Figure 2: In analogical classifiers, both situations and outcomes represent ratios.

When the case base is fully known, except for the outcome r_t of one case $c_t = (t, r_t)$, the transfer inference consists in finding the outcome r_t that minimizes the value of the Γ indicator:

$$r_t = \arg\min_{r \in \mathcal{O}} \Gamma(\sigma_S, \sigma_R, CB \cup \{(t, r)\})$$

4. Analogical Proportion-based Classification as a Case-Based Prediction Method

This section proposes to establish a correspondence between APC and case-based prediction, showing the former can be viewed as a special kind of the latter. This correspondence is illustrated by the diagram given in Fig. 2 that represents the APC in a similar view as case-based prediction, whose diagram is given in Fig. 1. More precisely, APC can be considered as applying a specific transfer by rule-based voting method: first, cases are *differences* between two instances (principle [P1]), and a single rule is triggered, that states that maximally similar situations (principle [P2]) should be associated with maximally similar outcomes. This section makes explicit all components of the case-based prediction configuration that can be associated to a given analogical proportion-based classifier, discussing successively the considered case base and similarity measures, as well as the applied transfer strategy.

4.1. Case Base

When seen as a case-based prediction method, APC works by comparing some *ratios* $\mathbf{a} : \mathbf{b}$ and $f(\mathbf{a}) : f(\mathbf{b})$ between the instances and their respective labels. Assuming that both instances and labels are vectors, these ratios are represented by the differences $s = \mathbf{a} - \mathbf{b}$ and $r_s = f(\mathbf{a}) - f(\mathbf{b})$ between two vectors. Let us denote by $x \in D$ a new instance for which the class $f(x)$ is to be predicted. Let C be the set of potential classes for $f(x)$, and $y \in C$. The source case c_s and potential new case $\hat{c}_t$ are of the following form:

$$\begin{aligned} c_s &= (\mathbf{a} - \mathbf{b}, f(\mathbf{a}) - f(\mathbf{b})) \\ \hat{c}_t &= (\mathbf{c} - x, f(\mathbf{c}) - y) \end{aligned} \tag{1}$$

where $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ are instances of D, and $f(\mathbf{a})$, $f(\mathbf{b})$, $f(\mathbf{c})$ their associated classes, represented as one-hot encoding vectors. APC implements the [P1] principle: cases are *differences* between instances of the data set D.

4.2. Similarity Measures

The two similarity measures σ_S and σ_R are constructed from the analogical dissimilarity AD, by noticing that AD measures a distance $AD(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) = \delta(\mathbf{a}-\mathbf{b}, \mathbf{c}-\mathbf{d})$ between two differences $\mathbf{a}-\mathbf{b}$ and $\mathbf{c}-\mathbf{d}$. The similarity measures σ_S and σ_R are obtained by applying a strictly decreasing function to the distance δ, e.g., by choosing $\sigma_S = \sigma_R = e^{-\delta}$. The similarity measure σ_S is such that the four instances $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ form an analogical proportion iff $\sigma_S(\mathbf{a}-\mathbf{b}, \mathbf{c}-\mathbf{d}) = 1$. The similarity measure σ_R is such that the four instances $f(\mathbf{a}), f(\mathbf{b}), f(\mathbf{c}), f(\mathbf{d})$ form an analogical proportion iff $\sigma_R(f(\mathbf{a}) - f(\mathbf{b}), f(\mathbf{c}) - f(\mathbf{d})) = 1$.

4.3. Transfer Strategy

When APC is viewed as a case-based prediction method, its transfer strategy is a rule-based voting strategy [35]. To see why, consider the decomposition described in [36] of the prediction procedure as an aggregation of the potential solutions y found for each instance $\mathbf{c} \in D$ followed by a majority vote. In this view, the search for potential solutions y consists in successively:

1. enumerating all instances $\mathbf{c}$, and for each one of them,
2. *Retrieval*: retrieve all source cases $c_s = (s, r_s) = (\mathbf{a} - \mathbf{b}, f(\mathbf{a}) - f(\mathbf{b}))$;
3. *Mapping*: compute the similarity $\sigma_S(s, t)$ between $s = \mathbf{a} - \mathbf{b}$ and $t = \mathbf{c} - x$;
4. *Transfer*: if $\sigma_S(s, t) = 1$ holds (*i.e.*, $\mathbf{a}, \mathbf{b}, \mathbf{c}, x$ are s.t. $\mathbf{a} : \mathbf{b} :: \mathbf{c} : x$), find the solutions y such that $\sigma_R(r_s, r) = 1$, with $r_s = f(\mathbf{a}) - f(\mathbf{b})$ and $r = f(\mathbf{c}) - y$.

This decision procedure thus considers all pairs $(c_s, \hat{c}_t)$ that can be obtained from a triple $(\mathbf{a}, \mathbf{b}, \mathbf{c})$, and searches for potential solutions y that can be inferred by applying the following similarity-based inference on a pair $(c_s, \hat{c}_t)$:

$$\frac{(\sigma_S = 1) \rightarrow (\sigma_R = 1) \quad \sigma_S(s, t) = 1}{\sigma_R(r_s, r) = 1}$$

The analogical root of x corresponds to the set of triples (**a**,**b**,**c**) for which the similarity-based inference allows to infer a solution y. The predicted solution $f(x)$ is the solution y that was inferred on the maximal number of pairs $(c_s, \hat{c}_t)$ by triggering the rule.

If the analogical root of x is empty, analogical classifiers extend the search to triples with lowest analogical dissimilarity, *i.e.*, with highest value for the similarity σ_S. This amounts to relaxing the condition $\sigma_S(s, t) = 1$ to the condition $\sigma_S(s, t) \approx 1$. The similarity-based inference becomes:

$$\frac{(\sigma_S = 1) \rightarrow (\sigma_R = 1) \quad \sigma_S(c_s, \hat{c}_t) \approx 1}{\sigma_R(c_s, \hat{c}_t) = 1}$$

Only the k solutions y that were derived from the rule $(\sigma_S = 1) \rightarrow (\sigma_R = 1)$ with the highest values of $\sigma_S(c_s, \hat{c}_t)$ are added to the solution set. Therefore, when viewed as case-based prediction methods, analogical proportion-based classifiers implement the relaxed form of the [P2] principle (only most similar cases are compared).

Algorithm 1 CoAT-APC

inputs: D (data set), x (additional instance), C (set of potential classes for $f(x)$), σ_S, σ_R (situation and outcome similarity measures), k (number of neighbor instances), n (size of the case base).
output: the predicted value $f(x)$ for the new instance x
$\ell \leftarrow \{y : 0 \text{ for } y \in C\}$
$\mathcal{N}(x) \leftarrow$ the k instances $\mathbf{c} \in D$ that maximize $\sigma_S(\mathbf{c}, x)$
for $\mathbf{c} \in \mathcal{N}_k(x)$ **do**
  $t \leftarrow \mathbf{c} - x$
  $CB(\mathbf{c}) \leftarrow$ the n source cases $c_s = (s, r_s)$ that maximize $\sigma_S(s, t)$
  for all $y \in C$ **do**
    $r \leftarrow f(\mathbf{c}) - y$
    $\ell[y] \leftarrow \ell[y] + \Gamma(\sigma_S, \sigma_R, CB(\mathbf{c}) \cup \{(t, r)\}))$
  end for
end for
$f(x) \leftarrow \arg\min_{y \in C} \ell[y]$
return $f(x)$

5. Tweaking CoAT to be an Analogical Proportion-based Classifier

The CoAT case-based prediction algorithm is modified in order to implement the two key principles [P1] and [P2] in its relaxed form. The resulting analogical proportion-based classifier, called CoAT-APC, implements a transfer strategy based on the minimization of a dataset complexity measure instead of being rule-based. Algo. 1 provides the pseudo-code description of the resulting CoAT-APC algorithm.

5.1. Proposed CoAT-APC Algorithm

Regarding the case base, the source cases c_s and potential new cases $\hat{c}_t$ are defined as in the previous section (Eq. 1) as differences between instances of the data set (principle [P1]).

Regarding the transfer strategy, CoAT-APC selects a set of instances $\mathbf{c} \in D$, and for each of them applying the CoAT method to evaluate the plausibility of each potential outcome (*i.e.*, class difference) $r = f(\mathbf{c}) - y$ associated with the target situation $t = \mathbf{c} - x$. The plausibility estimations obtained for each $y \in C$ are then aggregated to propose a solution $f(x)$. The decision procedure thus consists in successively:

1. finding the k instances $\mathbf{c}$ that maximize $\sigma_S(\mathbf{c}, x)$, and for each one of them,
2. forming a case base $CB(\mathbf{c})$ with the n source cases $s_c = (s, r_s)$ that maximize $\sigma_S(s, t)$, with $s = \mathbf{a} - \mathbf{b}$ and $t = \mathbf{c} - x$ (principle relaxed [P2]),
3. computing and storing $\Gamma(\sigma_S, \sigma_R, CB(\mathbf{c}) \cup \{(t, r)\})$ formed with $t = \mathbf{c} - x$ and $r = f(\mathbf{c}) - y$, for each potential solution $y \in C$,

Type	Dataset	Instances	Features	Classes
Discrete	balance	625	4	3
	car	1728	6	4
	monks1	432	6	2
	monks2	432	6	2
	monks3	432	6	2
	spect	238	23	2
	voting	435	16	2
	breastw	682	10	2
Continuous	iris	150	4	3
	pima	768	8	2
	user	258	5	4
	wine	178	4	3

Table 1
Data sets used for the experimental study of the proposed algorithm CoAT-APC and its variants.

4. aggregating the plausibility estimations by summing over **c**:

$$f(x) = \arg\min_{y \in C} \sum_{\mathbf{c}} \Gamma(\sigma_S, \sigma_R, CB(\mathbf{c}) \cup \{(\mathbf{c} - x, f(\mathbf{c}) - y)\})$$

5.2. Computational Complexity Analysis

The first step (finding the k instances **c** that maximize $\sigma_S(\mathbf{c}, x)$) is in line with [11]. It was shown experimentally to slightly improve the results while greatly reducing the computational cost of the decision procedure. The second step (forming the case base $CB(\mathbf{c})$) requires to precompute and store all $|D|^2$ differences between instances of D (which is done beforehand), and then sort these differences at runtime by decreasing value of $\sigma_S(s, t)$. The sorting procedure is done in $\mathcal{O}(|D|^2 \log(|D|))$, which may be the most costly part of the algorithm. The third step (computing Γ for each potential $y \in C$) can be done in $\mathcal{O}(n^2|C|)$, as shown in [9], which is tractable since optimal results are usually obtained with $n \leq 40$, as experimental results show. The last step consists in summing the plausibility estimations over **c** for each potential solution $y \in C$, and selecting the one that minimizes the sum. The overall computational complexity of the method is $\mathcal{O}(k \times (|D|^2 \log(|D|) + n^2|C|))$.

6. Experiments

This section describes the experiments run to validate the proposed CoAT-APC algorithm and the strategy it relies on, and to determine the impact that the principles [P1] and [P2] have on the performance of the proposed case-based prediction instantiation of analogical proportion-based classification.

Algorithm 2 CoAT

inputs: D (data set), x (additional instance), C (set of potential classes for $f(x)$), σ_S, σ_R (situation and outcome similarity measures).
output: the predicted value $f(x)$ for the new instance x
$CB \leftarrow D$
$f(x) \leftarrow \arg\min_{y \in C} \Gamma(\sigma_S, \sigma_R, CB \cup \{(x, y)\}))$
return $f(x)$

6.1. Experimental Protocol

The data sets used for classification are taken from the UCI repository[1], their characteristics are summarized in Table 1. They include 8 data sets with only nominal features and 4 data sets with only numerical features, in both cases associated with classification tasks with 2 to 4 classes. In all experiments, the two similarity measures σ_S and σ_R are fixed:

- $\sigma_S = e^{-ED}$, where $ED = \|\cdot\|_2$ is the standard Euclidean distance;
- $\sigma_R(u, v) = 1$ if $u = v$, and 0 otherwise.

Four algorithms are considered for comparison, applied to each data set D:

- CoAT: the case-based prediction algorithm, as of [9].The source cases are the instances of D, and the case base CB contains the whole data set D (see Algorithm 2).
- CoAT+[P1]: a modification of CoAT that implements principle [P1]. The source cases $c_s = (\mathbf{a} - \mathbf{b}, f(\mathbf{a}) - f(\mathbf{b}))$ are differences between instances of the data set D. The algorithm is the same as the one of CoAT-APC, but for each instance $\mathbf{c}$, the case base $CB(\mathbf{c})$ includes n randomly chosen source cases.
- CoAT+[P2]: a modification of CoAT that implements the relaxed form of principle [P2]. The source cases are instances of D, but the case base CB contains only the n instances $s \in D$ that maximize $\sigma_S(s, x)$ (see Algorithm 3).
- CoAT-APC: a combination of the two previous approaches. The source cases $c_s = (\mathbf{a} - \mathbf{b}, f(\mathbf{a}) - f(\mathbf{b}))$ are differences between instances of the data set D, and the case base $CB(\mathbf{c})$ contains the n source cases that maximize $\sigma_S(\mathbf{a} - \mathbf{b}, \mathbf{c} - x)$.

For each task, the performance is measured by the prediction accuracy, with 10-fold cross validation. The algorithm CoAT+[P2] is tested on each data set with a parameter n (the size of the case base CB) varying between 5 and $|D|$, by steps of 5. The algorithms CoAT+[P1] and CoAT-APC are tested on each data set for all pairs (k, n) with k varying between 3 and 51, by steps of 2, and n varying between 5 and 50 by steps of 5.

6.2. Results

Table 2 gives the classification results. For each data set, the best results considering standard deviations are marked in bold. When the two principles [P1] and [P2] are combined (algorithm

[1] https://archive.ics.uci.edu/ml/

Algorithm 3 CoAT+[P2]

inputs: D (data set), x (additional instance), C (set of potential classes for $f(x)$), σ_S, σ_R (situation and outcome similarity measures), n (size of the case base).
output: the predicted value $f(x)$ for the new instance x
$CB \leftarrow$ the n instances $c_s = (s, f(s)) \in D$ that maximize $\sigma_S(s, x)$
$f(x) \leftarrow \arg\min_{y \in C} \Gamma(\sigma_S, \sigma_R, CB \cup \{(x, y)\})$
return $f(x)$

Dataset	CoAT	CoAT+[P1]			CoAT+[P2]		CoAT-APC		
			k	n		n		k	n
balance	61.0% ± 5.45	77.1% ± 3.76	5	45	**92.8%**±2.44	180	**93.8%**±2.78	39	20
car	62.5% ± 2.51	81.3% ± 1.49	19	10	82.7% ± 2.61	85	**96.5%**±0.97	11	5
monks1	66.6% ± 6.85	91.9% ± 3.20	7	30	88.3% ± 5.38	20	**100%** ± 0.00	7	5
monks2	65.7% ± 0.35	74.4% ± 6.24	3	40	65.2% ± 1.45	55	**91.1%**±4.99	25	20
monks3	82.1% ± 7.09	**98.7%**±1.61	9	35	**97.1%**±2.57	45	**98.9%**±1.19	7	20
spect	**79.3%**±1.50	**82.7%**±9.46	11	15	**84.2%**±5.51	90	**82.0%**±6.99	47	35
voting	88.0% ± 3.96	**92.7%**±3.47	7	15	**94.0%**±3.24	65	**95.8%**±2.51	7	20
breastw	65.0% ± 0.33	67.1% ± 3.09	25	35	66.6% ± 1.88	400	**96.6%**±1.97	21	10
iris	**95.3%**±5.20	**98.6%**±2.67	17	40	**96.7%**±3.33	15	**99.3%**±2.00	27	20
pima	65.1% ± 3.42	**75.5%**±3.51	15	30	**75.4%**±3.91	270	**75.3%**±3.47	27	10
user	63.6% ± 4.59	31.9% ± 5.23	3	5	71.5% ± 6.55	25	**97.0%**±1.48	23	15
wine	64.9% ± 7.59	22.0% ± 9.47	3	10	60.6% ± 5.88	90	**94.4%**±4.31	51	5

Table 2
Classification results.

CoAT-APC), the resulting system offers a very good performance, and gives the best results for all data sets. This allows to validate the proposed approach and the integration of the analogical proportion principles into case-based predictions. When applied independently, the design principles [P1] and [P2] seem to have different impacts on the performance. Running CoAT with a case base restricted to the n cases that are most similar with the target situation (algorithm CoAT+[P2]) generally improves the performance, but not for all data sets (see e.g., breastw, wine, or monks2). Working on cases defined as differences between instances of D but with a random case base (algorithm CoAT+[P1]) often improves the performance as well, but for some data sets the performance results are surprisingly low (see e.g., user, wine, breastw, or even monks2). All tests were run using a fixed similarity measure σ_S, based on the Euclidean distance, which may not be optimal for all data sets. This may explain why the CoAT algorithm sometimes gives rather poor results. However, applying [P1] and [P2] design principles (CoAT-APC) greatly improves the performance of the classifier, even with this non-optimal similarity measure. In addition, it can be observed that the CoAT-APC algorithm obtains the best results for fairly low values of n, usually lower than 20. This parameter determines the size of the case base $CB(\mathbf{c})$. The computing time remains moderate: for the Balance Scale data set, with $k = 39$ and $n = 20$, predicting the class of a new instance takes 11.5 seconds on a current PC.

7. Conclusion and Future Works

This paper proposes an approach to bridge the gap between analogical proportion-based classifiers and case-based prediction algorithms, by showing that analogical proportion-based classification can be interpreted as a special kind of case-based prediction algorithm in which cases are differences between two instances of the data set, and maximally similar cases are compared to predict the class of a new instance. Results show that if these two design principles taken independently have an impact on the prediction performance of the case-based prediction system, they are especially powerful when combined, even when the prediction is done with a non-optimal similarity measure. On the contrary, the rule-based transfer strategy of analogical proportion-based classifiers seems to have a little impact on their efficiency: replacing it with a different transfer strategy, such as the CoAT's optimization strategy, leads to excellent performance results.

Future works will include comparing the CoAT-APC algorithm with state-of-the-art analogical proportion-based classification algorithms, both in terms of performance and computing time. More generally, there is a need for a shared implementation of the main case-based prediction algorithms, so that their performance can be compared on controlled benchmarks. The study also shows that as a case-based prediction algorithm, analogical proportion-based classifiers use a similarity measure constructed from the Euclidean distance. Future works will include learning a similarity measure that is more adequate to each considered case-based prediction task.

References

[1] E. Hüllermeier, Case-Based Approximate Reasoning, Theory and Decision Library B, Springer, 2007.

[2] D. Dubois, E. Hüllermeier, H. Prade, Fuzzy methods for case-based recommendation and decision support, Journal of Intelligent Information Systems 27 (2006) 95–115.

[3] L. Miclet, S. Bayoudh, A. Delhay, Analogical dissimilarity: Definition, algorithms and two experiments in machine learning, Journal of Artificial Intelligence Research 32 (2008) 793–824.

[4] N. Hug, H. Prade, G. Richard, M. Serrurier, Analogical classifiers: A theoretical perspective, in: Proc. of the 22nd European Conf. on Artificial Intelligence, ECAI, volume 285, 2016, pp. 689–697.

[5] N. Hug, H. Prade, G. Richard, M. Serrurier, Analogical proportion-based methods for recommendation – First investigations, Fuzzy Sets and Systems 366 (2019) 110–132.

[6] M. Bounhas, Logical Proportions for Classification and Preference Learning, HDR, Tunis University, 2021.

[7] H. Prade, G. Richard, Analogical proportions: why they are useful in AI, in: Proc. of the Int. Joint Conf. on Artificial Intelligence, IJCAI, 2021, pp. 4568–4576.

[8] F. Badra, A Dataset Complexity Measure for Analogical Transfer, in: Proc. of the Int. Joint Conf. on Artificial Intelligence, IJCAI, 2020, pp. 1601–1607.

[9] F. Badra, M.-J. Lesot, Theoretical and experimental study of a complexity measure for analogical transfer, in: Proc. of the Int. Conf. on Case-based Reasoning ICCBR, 2022.
[10] S. Bayoudh, Learning by analogy: a classification rule for binary and nominal data, in: Proc. of the Int. Joint Conf. on Artificial Intelligence, IJCAI, 2007, pp. 678–683.
[11] M. Bounhas, H. Prade, G. Richard, Analogy-based classifiers for nominal or numerical data, Int. J. of Approximate Reasoning 91 (2017) 36–55.
[12] M. Couceiro, N. Hug, H. Prade, G. Richard, Analogy-preserving functions: A way to extend Boolean samples, in: Proc. of the Int. Joint Conf. on Artificial Intelligence, IJCAI, 2017, pp. 1575–1581.
[13] D. R. Hofstadter, E. Sander, Surfaces and Essences: Analogy as the Fuel and Fire of Thinking, Basic Books, 2013.
[14] H. Gust, U. Krumnack, K. Kühnberger, A. Schwering, Analogical reasoning: A core of cognition., KI - Künstliche Intelligenz 22 (2008) 8–12.
[15] M. d'Aquin, F. Badra, S. Lafrogne, J. Lieber, A. Napoli, L. Szathmary, Case base mining for adaptation knowledge acquisition, in: Proc. of the Int. Joint Conf. on Artificial Intelligence, IJCAI, 2007, pp. 750–755.
[16] F. Badra, R. Bendaoud, R. Bentebibel, P.-A. Champin, J. Cojan, A. Cordier, S. Desprès, S. Jean-Daubias, J. Lieber, T. Meilender, A. Mille, E. Nauer, A. Napoli, Y. Toussaint, TAAABLE: Text mining, ontology engineering and hierarchical classification for textual case-based cooking, in: Workshop Proc. of the 9th European Conf. on Case-Based Reasoning, 2008, pp. 219–228.
[17] S. Ontañón, E. Plaza, On knowledge transfer in case-based inference, in: ICCBR Proceedings, Springer, 2012, pp. 312–326.
[18] V. Jalali, D. Leake, N. Forouzandehmehr, Learning and applying case adaptation rules for classification: an ensemble approach, in: Proc. of the Int. Joint Conf. on Artificial Intelligence, IJCAI, 2017, pp. 4874–4878.
[19] B. Fuchs, J. Lieber, A. Mille, A. Napoli, Differential adaptation: An operational approach to adaptation for solving numerical problems with CBR, Knowledge-Based Systems 68 (2014) 103–114.
[20] F. Badra, Reasoning with co-variations, in: Proc. of the 17th Int. Conf. on Artificial Intelligence: Methodology, Systems and Applications, AIMSA, 2016.
[21] B. Bouchon-Meunier, L. Valverde, A fuzzy approach to analogical reasoning, Soft Computing 3 (1999) 141–147.
[22] V. V. Cross, T. A. Sudkamp, Similarity and compatibility in fuzzy set theory, volume 93 of *Studies in Fuzziness and soft computing*, Springer, 2002.
[23] F. Badra, K. Sedki, A. Ugon, On the role of similarity in analogical transfer, in: Case-Based Reasoning Research and Development, volume 11156, Springer, 2018, pp. 499–514.
[24] J. Lieber, M. d'Aquin, F. Badra, A. Napoli, Modeling adaptation of breast cancer treatment decision protocols in the Kasimir project, Applied Intelligence 28 (2008) 261–274.
[25] F. Badra, A. Cordier, J. Lieber, Opportunistic adaptation knowledge discovery, in: Case-Based Reasoning Research and Development, Springer, 2009, pp. 60–74.
[26] K. Hanney, M. T. Keane, The adaptation knowledge bottleneck: How to ease it by learning from cases, in: Case-Based Reasoning Research and Development, volume 1266, Springer, 1997, pp. 359–370.

[27] S. Craw, N. Wiratunga, R. C. Rowe, Learning adaptation knowledge to improve case-based reasoning, Artificial Intelligence 170 (2006) 1175–1192.
[28] J. Beringer, E. Hüllermeier, Case-based learning in a bipolar possibilistic framework, International Journal of intelligent Systems 23 (2008) 1119–1134.
[29] E. Hüllermeier, Credible case-based inference using similarity profiles, IEEE Transactions on Knowledge and Data Engineering 19 (2007) 847–858.
[30] E. Hüllermeier, D. Dubois, H. Prade, Fuzzy Rules in Case-Based Reasoning, in: In Proc. of the Conf. AFIA99 Raisonnement à partir de cas, 1999, pp. 45–54.
[31] E. Hüllermeier, D. Dubois, H. Prade, Knowledge-based extrapolation of cases: a possibilistic approach, in: Technologies for Constructing Intelligent Systems 1, volume 89, Physica-Verlag HD, 2002, pp. 377–390.
[32] E. Hüllermeier, Case-Based Approximate Reasoning, Springer, 2007.
[33] D. Dubois, E. Hüllermeier, H. Prade, Flexible control of case-based prediction in the framework of possibility theory, in: EWCBR, Springer, 2000, pp. 61–73.
[34] E. Hüllermeier, Possibilistic instance-based learning, Artificial Intelligence 148 (2003) 335–383.
[35] M. Bounhas, H. Prade, G. Richard, Analogical classification: a rule-based view, in: Proc. of the Int. Conf. o Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU, volume 443, Springer, 2014, pp. 485–495.
[36] J. Lieber, E. Nauer, H. Prade, When revision-based case adaptation meets analogical extrapolation, in: Case-Based Reasoning Research and Development, volume 12877, Springer, 2021, pp. 156–170.

Interactions Between Knowledge Graph-Related Tasks and Analogical Reasoning: A Discussion

Pierre Monnin[1,*], Miguel Couceiro[2]

[1]Orange, Belfort, France
[2]Université de Lorraine, CNRS, LORIA, Nancy, France

Abstract

Analogical reasoning has been extensively studied and relies on statements of the form "A is to B as C is to D" that are called analogical proportions. The motivation of our work is based on the following twofold observation. On the one hand, recent analogy-based settings relying on character or word embeddings have achieved state-of-the-art performance on Natural Language Processing tasks. On the other hand, graph embedding approaches are now mainstream for knowledge graph-related tasks, *e.g.*, knowledge discovery, knowledge graph refinement, or recommendation. Inspired by these works, we advocate for the further study of interactions between knowledge graph-related tasks and analogical reasoning. In particular, we outline how knowledge graph embeddings combined with analogical reasoning could support semantic table interpretation, knowledge matching, and recommendation.

Keywords

Analogical reasoning, Graph Embedding, Semantic Table Interpretation, Knowledge Matching, Recommendation

1. Introduction

Analogical reasoning is a remarkable capability of the human mind [1]. *Analogical proportions* or, simply, *analogies*, are statements of the form "A is to B as C is to D"' which are often written as $A : B :: C : D$. A typical example of an analogy would be "Paris is to France as Stockholm is to Sweden". Most of the recent works on analogy use the formalization proposed in Lepage [2], and that subsumes common intuition on analogies viewed as a geometric proportion (Equation (1)), an arithmetic proportion (Equation (2)), or as a parallelogram in a vector space (Equation (3)):

$$\frac{A}{B} = \frac{C}{D} \quad (1) \qquad A - B = C - D \quad (2) \qquad \overrightarrow{A} - \overrightarrow{B} = \overrightarrow{C} - \overrightarrow{D} \quad (3)$$

Traditional tasks related to analogical reasoning include analogy detection (*i.e.*, classifying a quadruple as a valid or invalid analogy) and analogy solving (*i.e.*, finding an x such that

ICCBR Analogies'22: Workshop on Analogies: from Theory to Applications at ICCBR-2022, September, 2022, Nancy, France

*Corresponding author.
pierre.monnin@orange.com (P. Monnin); miguel.couceiro@loria.fr (M. Couceiro)
https://pmonnin.github.io (P. Monnin); https://members.loria.fr/mcouceiro/ (M. Couceiro)
0000-0002-2017-8426 (P. Monnin); 0000-0003-2316-7623 (M. Couceiro)

$A : B :: C : x$ constitutes a valid analogy). Analogies have been extensively studied in Natural Language Processing settings with applications in word morphology [3, 4], machine translation [5] and semantic tasks [6, 7, 8].

Also, knowledge graphs (KGs) have gained a significant interest from both academic and industrial actors. A KG can be seen defined as "a *graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities*" [9]. Atomic elements of KGs are *triples* $\langle s, p, o \rangle$ where s is the subject, p the predicate, and o the object of the triple respectively. An example of a triple could be $\langle \texttt{Paris}, \texttt{capitalOf}, \texttt{France} \rangle$, where the predicate (also called property) `capitalOf` qualifies the relation holding between `Paris` and `France`. KGs support several downstream applications including offering a consolidated view of knowledge scattered across sources, fact-checking, search engines, e-commerce, question answering, or recommendation [10, 11, 12, 13, 14]. Various techniques have been developed to build, refine, and use KGs, including Knowledge Graph Embedding (KGE) techniques which have shown impressive performance [12, 15]. Interestingly, the parallelogram view of an analogy (Equation (3)) can be related to the translational view adopted by some KGE models. For example, TransE [16] models a triple $\langle \texttt{Paris}, \texttt{capitalOf}, \texttt{France} \rangle$ as a translation $\overrightarrow{\texttt{Paris}} + \overrightarrow{\texttt{capitalOf}} = \overrightarrow{\texttt{France}}$. Hence, we would have:

$$\overrightarrow{\texttt{France}} - \overrightarrow{\texttt{Paris}} = \overrightarrow{\texttt{Sweden}} - \overrightarrow{\texttt{Stockholm}} = \overrightarrow{\texttt{capitalOf}}$$

It is noteworthy that some embedding techniques already consider analogical properties. For example, Liu et al. [17] argue that analogical inference is desirable for knowledge graph completion and include analogical structures in their learning objective. Alternatively, Portisch et al. [18] evaluate link prediction and data mining approaches developed for knowledge graphs on an analogy inference task with the goal of retrieving the last element (D) of a quadruple given the three first elements (A, B, and C). Inspired by such previous work, we advocate in this article for a further study of interactions between analogical reasoning and knowledge graph-related tasks.

This paper is organized as follows. In Section 2, we discuss possible interactions between analogical reasoning and Semantic Table Interpretation (STI) as STI can be supported by knowledge graph embeddings. In Section 3 we reformulate knowledge matching in terms of analogical proportions, and we further explore this discussion for knowledge graph-based recommendation (Section 4). We then conclude by briefly outlining some noteworthy perspectives in Section 5.

2. Analogies for Semantic Table Interpretation

Semantic Table Interpretation (STI) aims at understanding the semantic content of tabular data such as Excel or CSV files, or Web tables. This process is carried out by annotating elements of tables with constituents of a knowledge graph through the three following tasks:

Cell-Entity Annotation (CEA) associates cells with entities;

Column-Type Annotation (CTA) associates columns with types;

Columns-Property Annotation (CPA) associates pairs of columns with properties.

Table 1
Example of a table listing countries, their capitals, their official language(s), and their GDP. This table is inspired from the Wikipedia pages "List of countries and dependencies and their capitals in native languages"[1] and "List of countries by GDP (nominal)".[2]

Country	Capital	Official language(s)	GDP (US$ million)
Finland	*(empty)*	Finnish, Swedish	297,617
France	Paris	French	2,936,702
Germany	Berlin	German	4,256,540
Sweden	Stockholm	Swedish	621,241
Switzerland	Bern (de facto)	German, French, Italian, Romansh	841,969

STI has seen a growing research interest over the past few years, for example with the SemTab challenge [19]. Indeed, large parts of company knowledge or knowledge available on the Web are encoded as tabular data. Consequently, understanding the content of tables paves the way for several downstream tasks such as table completion with KG content, KG completion with table content, or data set search services [20].

When interpreting tabular data, several issues arise, *e.g.*, different encoding charsets, misaligned cells, or missing values (for example, the capital of Finland in Table 1). Tables alone also provide little context to help disambiguate candidate entities for cell annotation [21]. For example, consider Table 1 and its annotation with Wikidata, an encyclopedic knowledge graph [22]. Based solely on entity labels and string matching, annotation candidates for cell "Germany" are entity Q142[3] (Germany, the European country) and entity Q1350565[4] (Germany, the constituency of the European Parliament). To cope with such issues, current STI approaches rely on syntactic lookups and majority voting [23, 24], or graph embedding-based disambiguation [25]. In the latter case, Chabot et al. [25] rely on the assumption that columns of tables are semantically coherent. Thus, when applying a clustering algorithm on the embeddings of candidate entities for a whole column, valid entities should be grouped in the same cluster. In our example, Q142 should be grouped in the same cluster as the entities representing the other countries appearing in the table.

Interestingly, the semantic coherence of columns also allows to see a table through the lens of analogies. A first view consists in considering cells in pairs of columns as taking part in analogical proportions. For example, Table 1 can be seen as sets of analogies of the form `France : Paris :: Germany : Berlin` or `France : French :: Germany : German`. In such a setting, the task of filling missing table values can be thought of as an analogy solving task, *e.g.*, we would like to find x such that `France : Paris :: Finland : x` is a valid analogy. In STI, such as task could be carried out both by retrieval (in case the correct entity is in the knowledge graph) and generation (in case the correct entity is absent from the target knowledge graph). Regarding disambiguation between candidate entities, this could be achieved by choosing the entity that satisfies the highest number of analogies generated from the table. However, it is

[2]https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_and_their_capitals_in_native_languages
[2]https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)
[3]https://www.wikidata.org/wiki/Q183
[4]https://www.wikidata.org/wiki/Q1350565

noteworthy that tables can lead to a high number of analogies. For example, only considering columns "Country" and "Capital" of Table 1 already produces 12 analogies. One could thus wonder about the computational complexity of such an approach. Future work could investigate the need for generating all possible analogies or, on the contrary, for restricting to the most useful analogies to the task at hand. Such a notion of usefulness may be task- or domain-dependent and remains to be defined and discussed. A first approach to generating all analogies or pruning redundant analogies can be achieved by taking into account properties such as the symmetry of analogical proportions (*i.e.*, $A : B :: C : D \rightarrow C : D :: A : B$).

Alternatively to generating analogies from pairs of columns independently, tables could be considered as whole in an analogical setting that follows the work of Prade and Richard [26] and Hug et al. [27]. Rows r_1, r_2, r_3, and r_4 could be seen as vectors $\overrightarrow{r_i} = (r_{i1}, r_{i2}, \ldots, r_{in})$ such that analogical proportions hold on some of their components $J \subset [1, n]$. Then, from the analogical inference principle, it follows that analogical proportions should also hold on the remaining components:

$$\frac{\forall j \in J, r_{1j} : r_{2j} :: r_{3j} : r_{4j}}{\forall k \in [1, n] \setminus J, r_{1k} : r_{2k} :: r_{3k} : r_{4k}} \tag{4}$$

This more holistic view may guide the STI process by focusing on analogical proportions that are valid on a high number of columns. However, in both views, analogical validity may not be possible over the entire table, *i.e.*, all generated analogies may not be detected as valid. In such case, analogical validity ratios may be interesting metrics to guide and evaluate the quality of the STI process.

Inspired by recent approaches [3, 17, 18], we assume that analogical reasoning for Semantic Table Interpretation could be supported by graph or table embeddings [12, 28]. However, some challenges inherent to tabular data must be integrated into analogical formalizations. For example, tables can contain cells with multiple entities (*e.g.* "Finnish, Swedish" in Table 1) and columns can involve a mix of entities and literals (*e.g.*, column "GDP (US$ million)"). This leads to consider multi-modal embeddings. In a table-graph multi-modal embedding space, one could also envision the CEA task as detecting or solving analogies of the form $r_{i1} : e_{i1} :: r_{i2} : e_{i2}$ where r_{ij} are cells of a table and e_{ij} are their matching entities in the knowledge graph.

3. Analogies for Knowledge Matching

Knowledge graphs are freely aggregated, published, and edited in the Web of data, and may thus overlap. Hence, a key task resides in matching (or aligning) their content [29]. This task encompasses the identification, within an aggregated knowledge graph or across knowledge graphs, of nodes that are equivalent, more specific, weakly related, or that represent contradictory knowledge units. Matching allows to obtain a consolidated view of scattered elements of knowledge which is beneficial to many applications, such as fact-checking or query answering. The task of matching elements of knowledge graphs has been extensively studied in the literature. We refer the interested reader to the book of Euzenat and Shvaiko [29] for a comprehensive review of existing work.

A knowledge matching task can be approached as an analogical setting. Indeed, nodes of knowledge graphs can be seen in analogical proportions with their neighbors. For ex-

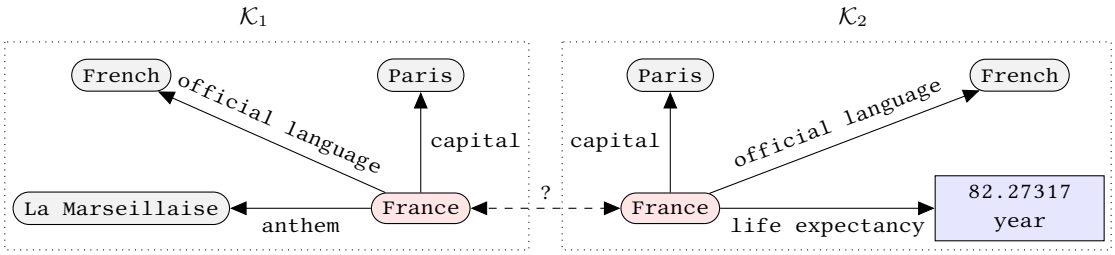

Figure 1: Example of a knowledge matching setting between two knowledge graphs $\mathcal{K}_1$ and $\mathcal{K}_2$ inspired from Wikidata.

ample, from the two knowledge graphs represented in Figure 1, it is possible to generate the analogy $\texttt{France}_{\mathcal{K}_1} : \texttt{Paris}_{\mathcal{K}_1} :: \texttt{France}_{\mathcal{K}_2} : \texttt{Paris}_{\mathcal{K}_2}$. The matching task then comes down to aligning nodes that maximize the validity of such analogical proportions between their respective neighbors with an analogy detection task. This corresponds to a structure-based matching [29]. This analogy-based matching process could be strengthen by considering existing alignments between neighbors (*e.g.*, $\texttt{Paris}_{\mathcal{K}_1}$ and $\texttt{Paris}_{\mathcal{K}_2}$) that could result from different matching methods (*e.g.*, string matching). For example, from the reflexivity property of analogical proportions (*i.e.*, $A : B :: A : B$), the inner symmetry (*i.e.*, $A : B :: C : D \implies B : A :: D : C$), the uniqueness postulate (*i.e.*, given A, B, and C, there exists only one D such that $A : B :: C : D$), the alignment $\texttt{Paris}_{\mathcal{K}_1} = \texttt{Paris}_{\mathcal{K}_2}$, and the analogical proportion $\texttt{Paris} : \texttt{France}_{\mathcal{K}_1} :: \texttt{Paris} : \texttt{France}_{\mathcal{K}_2}$, it follows that $\texttt{France}_{\mathcal{K}_1} = \texttt{France}_{\mathcal{K}_2}$. Such an analogical matching process could also produce valid results without preexisting alignments by only taking into account structural similarities. Thus, it could be used to start a matching pipeline. Note that the previous analogical proportion relies on similarities between identical nodes to match. By generating analogies based on granularity differences or contradictions between nodes, we could output such different alignment types. From the previous observations, a challenge thus resides in having a set of preexisting alignments of different types that could guide the analogy-based matching towards specific types of alignments.

It should be noted that other analogy-based views to match nodes can be considered. For example, given a set of preexisting alignments, matching a node $\texttt{France}_{\mathcal{K}_1}$ can be seen as solving a set of analogical equations of the form

$$\texttt{Paris}_{\mathcal{K}_2} : \texttt{Paris}_{\mathcal{K}_2} :: \texttt{France}_{\mathcal{K}_1} : x$$
$$\texttt{French}_{\mathcal{K}_2} : \texttt{French}_{\mathcal{K}_2} :: \texttt{France}_{\mathcal{K}_1} : x$$

and chosing the entity that is mostly output as x. Analogies could also serve as a basis to align predicates (*e.g.*, $\texttt{capital}_{\mathcal{K}_1}$ and $\texttt{capital}_{\mathcal{K}_2}$). Indeed, if two predicate are identical, then analogical proportions should hold between the entities they link, *e.g.*,

$$\texttt{France}_{\mathcal{K}_1} : \texttt{Paris}_{\mathcal{K}_1} :: \texttt{France}_{\mathcal{K}_2} : \texttt{Paris}_{\mathcal{K}_2}$$
$$\texttt{Germany}_{\mathcal{K}_1} : \texttt{Berlin}_{\mathcal{K}_1} :: \texttt{Germany}_{\mathcal{K}_2} : \texttt{Berlin}_{\mathcal{K}_2}$$

$$\texttt{France}_{\mathcal{K}_1} : \texttt{Paris}_{\mathcal{K}_1} :: \texttt{Germany}_{\mathcal{K}_2} : \texttt{Berlin}_{\mathcal{K}_2}$$

Hence, the alignment of predicates could be carried out by matching predicates that have a high number of valid analogical proportions between the entities they respectively connect.

Recent matching approaches rely on graph embeddings [30, 31, 32, 33]. Hence, it could be of interest to use such graph embeddings in an analogical setting for matching. This could correspond to aligning the embedding spaces of the KGs to match [34]. Enforcing analogical properties in the training procedure similarly to Liu et al. [17] could also be tested to learn specific graph embeddings tailored for analogical reasoning. However, it should be noted that analogy-based approaches to knowledge matching need to cope with issues similar to those described in Section 2. Indeed, KGs mix entities and literals (*e.g.*, the life expectancy in $\mathcal{K}_2$), which may require the use of multi-modal embeddings. Additionally, KGs may be incomplete and two equivalent nodes may not be entirely comparable based on their neigbhors. For example, in Figure 1, $\texttt{France}_{\mathcal{K}_1}$ is associated with its anthem `La Marseillaise` which is absent from $\mathcal{K}_2$. Additionally, not all nodes from a KG may found their counterpart in another KG. Hence, an analogy-based matching approach should try to maximize analogical validity without reaching full coverage. Due to the increasing size of KGs, the computational complexity of such an analogy-based matching approach and the need for generating all possible analogies or only the most useful should also be taken into account.

4. Analogies for Knowledge Graph-Based Recommendation

In this section, we consider the task of recommending items to users. Traditional approaches rely on similarity between users and/or items. Indeed, collaborative filtering-based recommender systems simultaneously consider similarities between users, items, and users and items based on their interactions. Alternatively, content based-recommender systems consider features of items to find and recommend items similar to the ones liked by the users.

As such, recommendation is a natural setting for analogical reasoning since it is also based on similarities. That is why, analogies have already been applied to recommendation with the objective of predicting the rating of an item by a user based on ratings of other similar users [27, 35]. Precisely, consider four users a, b, c, and d such that for each item j commonly rated, the analogical proportion $r_{aj} : r_{bj} :: r_{cj} : r_{dj}$ holds, with r_{aj} the rating of user a for item j. From the analogical inference principle, it is possible to predict the rating r_{di} for an item i that has only been rated by a, b, and c by solving the analogical proportion $r_{ai} : r_{bi} :: r_{ci} : x$. This analogy-based setting has also been adapted to preference learning with the objective of learning to rank a set of objects [36, 37], and considered in case-based reasoning [38, 39].

Recently, KGs have been introduced in recommender systems as sources of side information [11, 13]. Indeed, KGs allow to represent relations between items and their attributes, between users and items, and any additional user information. Hence, KGs better capture mutual relations between these different entities. Such rich KGs and their advantages motivated the use of knowledge graph embeddings for recommendation [11, 13]. However, these embeddings models do not take into account potential analogical constraints holding between users and items. Hence, we propose to study how knowledge graph embeddings could be combined with analogical proportions for recommendation. Such proportions could involve

users and items to directly support the recommendation, *e.g.*, $\texttt{user}_1 : \texttt{item}_1 :: \texttt{user}_2 : \texttt{item}_2$. We could also envision user-only analogies $\texttt{user}_1 : \texttt{user}_2 :: \texttt{user}_3 : \texttt{user}_4$ allowing to find similar users that could then support the recommendation of an item. Item-attribute analogies $\texttt{item}_1 : \texttt{attribute}_2 :: \texttt{item}_3 : \texttt{attribute}_4$ could highlight similarities between items whereas user-attribute analogies $\texttt{user}_1 : \texttt{attribute}_2 :: \texttt{user}_3 : \texttt{attribute}_4$ could emphasize the importance of some attributes to users. Such analogical proportions could be used to enrich training data or to check outputs of models by ensuring a minimum level of valid analogies with the recommended item(s). Alternatively, such analogies could be directly integrated in the learning procedure of the graph embeddings, similarly to the work of Liu et al. [17].

5. Conclusion & Perspectives

In this article, we advocated for the deeper study of the interactions between analogical reasoning and knowledge graph-related tasks. On the one hand, one can profit from recent analogy-based settings with state-of-the-art results on various tasks such as in Natural Language Processing and decision making, that make use of suitable data representations (embeddings). On the other hand, approaches based on knowledge graph embeddings are now mainstream and achieve competitive results for several tasks associated with knowledge graphs.

Motivated by these developments, we illustrated how analogy-based settings emerge naturally in semantic table interpretation, knowledge matching, and recommendation. While they could be suitably supported by available table or graph embeddings, such settings pose several challenges and open questions that need to be addressed. In particular, it remains to assess whether analogical views of such tasks actually improve performance. Interestingly, aside performance, such an integration of analogical reasoning could pave the way towards additional interpretability and explainability of approaches as discussed by Hüllermeier [40]. This could, in turn, strengthen the line of research studying knowledge graphs as tools for explainable AI [41].

References

[1] M. Mitchell, Abstraction and analogy-making in artificial intelligence, Annals of the New York Academy of Sciences 1505 (2021) 79–101.

[2] Y. Lepage, De l'analogie rendant compte de la commutation en linguistique, 2003. URL: https://tel.archives-ouvertes.fr/tel-00004372.

[3] S. Alsaidi, A. Decker, P. Lay, E. Marquer, P. Murena, M. Couceiro, A neural approach for detecting morphological analogies, in: 8th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2021, Porto, Portugal, October 6-9, 2021, IEEE, 2021, pp. 1–10. doi:10.1109/DSAA53316.2021.9564186.

[4] P. Murena, M. Al-Ghossein, J. Dessalles, A. Cornuéjols, Solving analogies on words based on minimal complexity transformation, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, ijcai.org, 2020, pp. 1848–1854. doi:10.24963/ijcai.2020/256.

[5] V. Taillandier, L. Wang, Y. Lepage, Réseaux de neurones pour la résolution d'analogies entre phrases en traduction automatique par l'exemple (neural networks for the resolution of analogies between sentences in EBMT), 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 2 : Traitement Automatique des Langues Naturelle, Nancy, France, June 8-19, 2020, ATALA et AFCP, 2020, pp. 108–121. URL: https://aclanthology.org/2020.jeptalnrecital-taln.9/.

[6] S. D. Afantenos, T. Kunze, S. Lim, H. Prade, G. Richard, Analogies between sentences: Theoretical aspects - preliminary experiments, in: J. Vejnarová, N. Wilson (Eds.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 16th European Conference, ECSQARU 2021, Prague, Czech Republic, September 21-24, 2021, Proceedings, volume 12897 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 3–18.

[7] S. Lim, H. Prade, G. Richard, Classifying and completing word analogies by machine learning, Int. J. Approx. Reason. 132 (2021) 1–25.

[8] Y. Lepage, Analogies between short sentences: A semantico-formal approach, in: Z. Vetulani, P. Paroubek, M. Kubis (Eds.), Human Language Technology. Challenges for Computer Science and Linguistics - 9th Language and Technology Conference, LTC 2019, Poznan, Poland, May 17-19, 2019, Revised Selected Papers, volume 13212 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 163–179.

[9] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutiérrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge Graphs, Synthesis Lectures on Data, Semantics, and Knowledge, Morgan & Claypool Publishers, 2021. doi:10.2200/S01125ED1V01Y202109DSK022.

[10] X. L. Dong, Building a broad knowledge graph for products, in: 35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019, IEEE, 2019, p. 25. doi:10.1109/ICDE.2019.00010.

[11] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems, IEEE Transactions on Knowledge and Data Engineering (2020) 1–1. doi:10.1109/TKDE.2020.3028705.

[12] S. Ji, S. Pan, E. Cambria, P. Marttinen, P. S. Yu, A survey on knowledge graphs: Representation, acquisition, and applications, IEEE Transactions on Neural Networks and Learning Systems 33 (2022) 494–514. doi:10.1109/TNNLS.2021.3070843.

[13] C. Liu, L. Li, X. Yao, L. Tang, A survey of recommendation algorithms based on knowledge graph embedding, in: 2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI), 2019, pp. 168–171. doi:10.1109/CSEI47661.2019.8938875.

[14] N. F. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, J. Taylor, Industry-scale knowledge graphs: lessons and challenges, Communications of the ACM 62 (2019) 36–43. doi:10.1145/3331166.

[15] M. Nickel, K. Murphy, V. Tresp, E. Gabrilovich, A review of relational machine learning for knowledge graphs, Proceedings of the IEEE 104 (2016) 11–33. doi:10.1109/JPROC.2015.2483592.

[16] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, 2013, pp. 2787–2795. URL: https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html.

[17] H. Liu, Y. Wu, Y. Yang, Analogical inference for multi-relational embeddings, in: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of *Proceedings of Machine Learning Research*, PMLR, 2017, pp. 2168–2178. URL: http://proceedings.mlr.press/v70/liu17d.html.

[18] J. Portisch, N. Heist, H. Paulheim, Knowledge graph embedding for data mining vs. knowledge graph embedding for link prediction - two sides of the same coin?, Semantic Web 13 (2022) 399–422. doi:10.3233/SW-212892.

[19] E. Jiménez-Ruiz, V. Efthymiou, J. Chen, V. Cutrona, O. Hassanzadeh, J. Sequeda, K. Srinivas, N. Abdelmageed, M. Hulsebos, D. Oliveira, C. Pesquita (Eds.), Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 27, 2021, volume 3103 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022. URL: http://ceur-ws.org/Vol-3103.

[20] A. Chapman, E. Simperl, L. Koesten, G. Konstantinidis, L. Ibáñez, E. Kacprzak, P. Groth, Dataset search: a survey, VLDB Journal 29 (2020) 251–272.

[21] S. Zhang, K. Balog, Web table extraction, retrieval, and augmentation: A survey, ACM Transactions on Intelligent Systems and Technology 11 (2020) 13:1–13:35. doi:10.1145/3372117.

[22] D. Vrandecic, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Communications of the ACM 57 (2014) 78–85. doi:10.1145/2629489.

[23] M. Cremaschi, R. Avogadro, D. Chieregato, Mantistable: an automatic approach for the semantic table interpretation, in: Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching co-located with the 18th International Semantic Web Conference, SemTab@ISWC 2019, Auckland, New Zealand, October 30, 2019, volume 2553 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 15–24. URL: http://ceur-ws.org/Vol-2553/paper3.pdf.

[24] V. Huynh, J. Liu, Y. Chabot, F. Deuzé, T. Labbé, P. Monnin, R. Troncy, DAGOBAH: table and graph contexts for efficient semantic annotation of tabular data, in: Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 27, 2021, volume 3103 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 19–31. URL: http://ceur-ws.org/Vol-3103/paper2.pdf.

[25] Y. Chabot, T. Labbé, J. Liu, R. Troncy, DAGOBAH: an end-to-end context-free tabular data semantic annotation system, in: Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching co-located with the 18th International Semantic Web Conference, SemTab@ISWC 2019, Auckland, New Zealand, October 30, 2019, volume 2553 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 41–48. URL: http://ceur-ws.org/Vol-2553/paper6.pdf.

[26] H. Prade, G. Richard, Reasoning with logical proportions, in: Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010, AAAI Press, 2010. URL: http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1413.

[27] N. Hug, H. Prade, G. Richard, M. Serrurier, Analogy in recommendation. numerical vs. ordinal: A discussion, in: 2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016, Vancouver, BC, Canada, July 24-29, 2016, IEEE, 2016, pp. 2220–2226. doi:10.1109/FUZZ-IEEE.2016.7737969.

[28] X. Deng, H. Sun, A. Lees, Y. Wu, C. Yu, TURL: table understanding through representation learning, Proceedings of the VLDB Endowment 14 (2020) 307–319. URL: http://www.vldb.org/pvldb/vol14/p307-deng.pdf. doi:10.5555/3430915.3442430.

[29] J. Euzenat, P. Shvaiko, Ontology Matching, Second Edition, Springer, 2013.

[30] E. Jiménez-Ruiz, A. Agibetov, J. Chen, M. Samwald, V. Cross, Dividing the ontology alignment task with semantic embeddings and logic-based modules, in: ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), volume 325 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2020, pp. 784–791. doi:10.3233/FAIA200167.

[31] P. Monnin, C. Raïssi, A. Napoli, A. Coulet, Discovering alignment relations with graph convolutional networks: A biomedical case study, Semantic Web 13 (2022) 379–398. doi:10.3233/SW-210452.

[32] N. Pang, W. Zeng, J. Tang, Z. Tan, X. Zhao, Iterative entity alignment with improved neural attribute embedding, in: Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG2019) Co-located with the 16th Extended Semantic Web Conference 2019 (ESWC 2019), Portoroz, Slovenia, June 2, 2019, volume 2377 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 41–46.

[33] Z. Wang, Q. Lv, X. Lan, Y. Zhang, Cross-lingual knowledge graph alignment via graph convolutional networks, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, Association for Computational Linguistics, 2018, pp. 349–357. doi:10.18653/v1/d18-1032.

[34] R. Biswas, M. Alam, H. Sack, Is aligning embedding spaces a challenging task? a study on heterogeneous embedding alignment methods, 2020. URL: https://arxiv.org/abs/2002.09247. arXiv:2002.09247.

[35] N. Hug, H. Prade, G. Richard, Experimenting analogical reasoning in recommendation, in: Foundations of Intelligent Systems - 22nd International Symposium, ISMIS 2015, Lyon, France, October 21-23, 2015, Proceedings, volume 9384 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 69–78. doi:10.1007/978-3-319-25252-0_8.

[36] M. A. Fahandar, E. Hüllermeier, Learning to rank based on analogical reasoning, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, AAAI Press, 2018, pp. 2951–2958. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16830.

[37] M. A. Fahandar, E. Hüllermeier, Analogical embedding for analogy-based learning to rank, in: Advances in Intelligent Data Analysis XIX - 19th International Symposium on Intelligent Data Analysis, IDA 2021, Porto, Portugal, April 26-28, 2021, Proceedings, volume 12695 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 76–88. doi:10.1007/978-3-030-74251-5_7.

[38] M. M. Richter, R. O. Weber, Case-based reasoning, Springer, 2016.

[39] J. Lieber, E. Nauer, H. Prade, When Revision-Based Case Adaptation Meets Analogical Extrapolation, in: 29th ICCBR, volume 12877 of *LNCS*, 2021, pp. 156–170.

[40] E. Hüllermeier, Towards analogy-based explanations in machine learning, in: Modeling Decisions for Artificial Intelligence - 17th International Conference, MDAI 2020, Sant Cugat, Spain, September 2-4, 2020, Proceedings, volume 12256 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 205–217. doi:10.1007/978-3-030-57524-3_17.

[41] I. Tiddi, S. Schlobach, Knowledge graphs as tools for explainable machine learning: A survey, Artificial Intelligence 302 (2022) 103627. doi:10.1016/j.artint.2021.103627.

An analogy based framework for patient-stay identification in healthcare

Safa Alsaidi[1,2,*,†], Miguel Couceiro[3], Esteban Marquer[3], Sophie Quennelle[1,2,5], Anita Burgun[1,2,4,5], Nicolas Garcelon[1,2,4,5] and Adrien Coulet[1,2]

[1]*Inria Paris, F-75012 Paris, France*

[2]*Centre de Recherche des Cordeliers, Inserm, Université Paris Cité, Sorbonne Université, F-75006 Paris, France*

[3]*LORIA, CNRS, Universite de Lorraine, F-54000, France*

[4]*Imagine Institute, F-75015 Paris, France*

[5]*Service d'Informatique Biomédicale, Hôpital Necker-Enfants Malades, Assistance Publique - Hôpitaux de Paris, F-75015 Paris, France*

Abstract

Analogical proportions are statements of the form *"A is to B as C is to D"*. Analogies have been used in various reasoning and classification tasks, addressing different domains. Representation learning has enabled interesting progress in various analogy reasoning applications, where it focuses on the challenge of obtaining a vector representation of complex data. In the biomedical domain, representation learning has been adapted to patient data to solve various tasks such as predicting readmission, diagnosis, and length of stay. In this paper, we focus on the particular task of patient-stay identification, *i.e.*, does a hospital stay belong to a patient or not? This constitutes a building block for addressing key biomedical tasks such as patient matching and privacy preservation. We propose a prototypical architecture that combines patient-stay representation learning and the analogical reasoning framework. For evaluation, we constitute sets of analogies from real-word Electronic Health Records, where objects are patient-stay representations learned from the data. We enrich our analogies using analogical properties and use them to train a neural model to detect whether an analogy is valid. We define three first experimental setups to address our task, present our empirical results, and discuss further perspectives.

Keywords

analogy classification, patient matching, electronic health records, patient representation learning,

1. Introduction

An *analogical proportion*, or simply *analogy*, is a quaternary relation involving four objects A, B, C, and D that draws a parallel between the relation between A and B and the relation between C and D, and that supports analogical reasoning. There are two common tasks associated with

ICCBR Analogies'22: Workshop on Analogies: from Theory to Applications at ICCBR-2022, September, 2022, Nancy, France

*Corresponding author.

safa.alsaidi@inria.fr (S. Alsaidi); miguel.couceiro@loria.fr (M. Couceiro); esteban.marquer@loria.fr (E. Marquer); sophie.quennelle@inria.fr (S. Quennelle); anita.burgun@aphp.fr (A. Burgun); nicolas.garcelon@institutimagine.org (N. Garcelon); adrien.coulet@inria.fr (A. Coulet)

0000-0002-4132-1068 (S. Alsaidi); 0000-0003-2316-7623 (M. Couceiro); 0000-0003-2315-7732 (E. Marquer); 0000-0002-4782-6737 (S. Quennelle); 0000-0001-6855-4366 (A. Burgun); 0000-0002-3326-2811 (N. Garcelon); 0000-0002-1466-062X (A. Coulet)

analogies, namely, *analogy detection* and *analogy solving*. Analogy detection aims at deciding whether a quadruple $\langle A, B, C, D\rangle$ constitutes a valid analogy. Analogy solving aims at finding an x that makes $A : B :: C : x$ a valid analogy. Analogy reasoning has been applied to different Natural Language Processing (NLP) tasks such as mining paradigm tables in linguistics and image generation [1, 2].

Representation learning consists of learning low-dimension feature representations (*i.e.*, embeddings) from data. These embeddings, or *vector representations*, of objects (*i.e.*, words, images, characters, etc.) underpin much of modern machine learning and have demonstrated impressive performance on various downstream NLP tasks. For instance, Lim et al. [3] proposed a deep learning model to tackle analogies using semantic embeddings. Their architecture integrates the characteristics of analogies by design and relies heavily on pretrained GloVe embeddings [4]. These embeddings were not trained explicitly to find analogies; yet they were able to detect differences between objects. Hertzmann et al. [5] proposed an analogical framework to learn "image filters" between a pair of images to create an "analogous" filtered result on a third image. The generated image D should relate to C in the same way as B relates to A. Alsaidi et al. [6] developed a neural approach and used character-based embeddings to detect morphological analogies between words.

Analogies have not been sufficiently exploited in healthcare, which thus motivates our work. However, practitioners unconsciously use analogical reasoning (*i.e.*, medical reasoning) in their daily clinical practice to understand the possible causes for a disease diagnosis and prognosis by linking visible signs and symptoms that have been observed among different patients. In addition, several machine learning methods were applied to investigate analogies in healthcare. For instance, Casteleiro et al. [7] utilized analogies to infer disease treatments from statements extracted from text. In their work, they try to extract biomedical facts by analogical reasoning from embeddings. Dynomant et al. [8] used analogical proportions to compare embedding methods trained on a corpus of French health-related documents (*i.e.*, discharge summary, procedure reports, and prescriptions). Analogical proportions were applied on the embeddings of medical documents to verify if $(\vec{A} - \vec{B}) + \vec{C} \approx \vec{D}$, thus allowing to check whether the similarity between A and B is similar to the one between C and D. An example of an analogical proportion they obtain is "(*cardiology* - *heart*) + *lung* ≈ *pneumology*." Rather et al. [9] used analogical proportions to identify hidden or unknown biomedical knowledge from free text resources. They proposed analogical proportions of the form "*acetaminophen* is a type of *drug* as *diabetes* is a type of *disease*."

In this paper, we aim to explore how the analogy framework can help in solving tasks relevant to the healthcare domain. We propose two models that learn patient-stay representations (*i.e.*, learn a vector representation of all the patient EHR data collected during a single stay) to detect analogies in healthcare. To do so, we define two crucial steps that are (1) the learning of embeddings adapted to patient data, and (2) the definition of a neural network dedicated to learn formal properties of analogy. As for the network, we use the same model that was proposed by Lim et al. [3] for word semantics, and later adapted by Alsaidi et al. [6] by incorporating character-based embeddings for morphological analogies. We argue that the framework itself has the potential to be applied in a wide range of domains, and we propose to use it here for healthcare applications, namely, for the patient identification task we introduce below.

Electronic Health Records (EHRs) are real world healthcare data that have been used to

train predictive models (including neural network models) for different biomedical tasks, *e.g.*, predicting patient mortality, hospital readmission, length of stay, etc. These EHRs consist of clinical and administrative data collected during patient hospital stays in the form of both *structured* and *unstructured* data. Structured data generally includes diagnostic codes, lab tests, demographics, admission-related information, etc. It can be either static, *e.g.*, patient demographics, or temporal, *e.g.*, vital signs. Unstructured data includes various documents in natural language such as clinical notes, nursing reports, discharge summaries, lab reports, etc. For this work, we consider EHRs from the MIMIC-III (Medical Information Mart for Intensive Care, version 3) database [10] to learn patient representations (*i.e.*, patient embeddings) by converting patient data from the raw EHRs to embeddings that can be further processed. MIMIC-III is a free publicly available hospital database containing de-identified patient health data. This database has been widely used by researchers conducting data mining and machine learning studies applied to healthcare.

Several neural network architectures have been developed to represent biomedical data. For instance, Si et al. [11] adapted a multi-level CNN to learn patient representations from clinical notes through a multi-task learning framework to predict patient mortality and length of stay. Zhang et al. [12] used GRU-based RNN to capture relationships between clinical events and employed attention mechanism to learn a personalized representation to predict patient's future hospitalization using EHR data. Madhumita et al. [13] used a stacked denoising autoencoder and a paragraph vector model to learn generalized patient representations directly from clinical notes to predict patient mortality, primary diagnostic, procedural category, and patient gender. Zhang et al. [14] proposed two neural network architectures that enhance patient representation learning by combining sequential unstructured notes with structured data and evaluated these representations on 3 risk evaluation tasks (*i.e.*, in-hospital mortality, 30-day hospital readmission, and length of stay prediction). In our paper, we learn patient-stay representations and consider the task of patient-stay identification. We think that the tools that address this task will serve as building blocks for more complex and key biomedical tasks, such as patient matching and privacy preservation checking [15, 16].

In this paper, we particularly propose to tackle this task by relying on the detection of analogies in healthcare. In Section 2, we define the setting of analogy that we work on. The models we propose to detect analogies are described in Section 3, along with the procedures we use for data augmentation, training, and evaluation. In Section 4, we provide a description of the MIMIC-III dataset and detail how we build our experimental dataset. We present our experiments and report our results in Section 5. In Section 6, we discuss perspectives for future research.

The main contributions of this paper are the following:

- we propose an analogy based setting using patient-stay representations;
- we propose an embedding model to learn patient-stay representations;
- we display the performance of our classification model to detect analogies on patient-stay data.

2. Defining the task

As we defined previously, an analogy is a 4-ary relation written as $A : B :: C : D$ and expressed as "A is to B as C is to D". In this paper, we work on patient-stay analogies, *i.e.*, on analogies involving hospital stay. In our setting, A, B, C, and D represent patient-stay representations. We define an analogy based setting on patient-stay data that we refer to as *Identity setting*. For that, we consider patient-stay representations, which are vector representations of EHR data that belong to a single hospital stay. Based on the type of EHR data that we decide to include, our patient-stay representations can be made of a representation of either structured or unstructured data, or they can be made of the concatenation of both types of data. More details are provided in Section 5. For this setting, we propose to build analogies of the form:

$$s_{t_1}^{i_1} : s_{t_2}^{i_1} :: s_{t_3}^{i_2} : s_{t_4}^{i_2}$$

where s_t^i refers to the stay t of patient i. Here, pairs of the analogy quadruples are made of two random stays belonging to the same patient. Since there is no constraint on the order of stays, $s_{t_1}^{i_1}$ can happen before $s_{t_2}^{i_1}$ or the inverse. Note that i_1 and i_2 can be the same patient, and that t_1 and t_2, or t_3 and t_4, can represent the same time stamp. Furthermore, t_1 and t_3 or t_2 and t_4 can be the same when $i_1 = i_2$ (but not when $i_1 \neq i_2$). The *Identity* setting finds applications in several tasks relevant to biomedical informatics, including:

- data cleaning,
- data privacy related application,
- patient matching.

Data cleaning applications in the health domain involve repairing or removing patient health data that is inaccurate, incorrectly structured, duplicative, or incomplete. In data cleaning applications, we can associate an erroneously affected sample of data to the patient it belongs. Privacy related applications include verifying if patient data is de-identified, and whether it can be re-identified using different systems. Patient matching is defined as the identification and linking of one patient's data within and across health databases in order to obtain a comprehensive view of that patient's health care record [17]. In patient matching, we try to match patient-related information, either a single patient data (*e.g.*, a document) or full EHR data, that can coexist in one or several databases.

In this paper, we try to match *patient-stay representations* to the patient they belong to. We focus on the task of patient-stay identification, where we aim to determine if a particular hospital stay belongs to a certain patient. We address this task by learning a model to classify such quadruples into valid and invalid analogies. In this sense, we implement the task of analogy detection that aims to determine if a quadruple is a valid analogy. For our *Identity setting*, we define a *valid analogy* as a quadruple of four stays $(s_{t_1}^{i_1}, s_{t_2}^{i_1}, s_{t_3}^{i_2}, s_{t_4}^{i_2})$, where each pair of two stays belong to a single patient i_j; other forms of analogies are considered invalid.

3. Proposed Approach

Our model is made of two components: an embedding model and a classification model. The second takes as input patient-stay representations computed by the first (see Section 3.1).

Our embedding model is trained along with the classification model. We also detail the data augmentation procedure in Section 3.2, and describe the training and evaluation protocols that we followed in Section 3.3.

3.1. Embedding and Classification Models

The models described in this subsection are schematized in Figure 1.

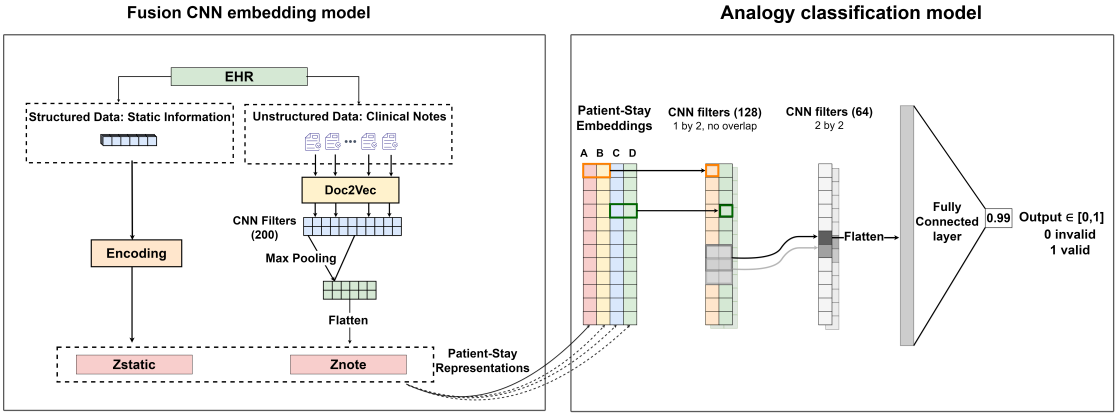

Figure 1: The Fusion CNN embedding model and the CNN classification model.

Embedding Model. As our embedding model, we adapt the Fusion CNN model that was developed by Zhang et al. [14] to obtain patient-stay representations. They proposed a neural network architecture that combines structured and unstructured data to obtain patient representations. The model consists of five parts: static information encoder, temporal signals embeddings, sequential notes representation, patient representation, and output layer that is used to predict three different clinical tasks.

In this work, we restricted structured data to static information, *i.e.*, demographics (Z_{demo}) and admission-related information (Z_{adm}), omitting deliberately vitals in this first attempt. Our model is thus made of static information encoder and sequential notes representation that are used to obtain patient-stay representations as illustrated by the left frame of Figure 1. The static categorical features are encoded as one-hot vectors through the static information encoder. The output of the encoder is $Z_{static} = [Z_{demo}; Z_{adm}]$, where $[Z_{demo}; Z_{adm}]$ is the concatenation of Z_{demo} and Z_{adm}. Z_{static} forms one part of the full patient-stay representation. As shown in Figure 1, the clinical notes representation part is made of a document embedding model, 2 convolutional layers, a max-pooling layer, and a flatten layer. To learn the document embeddings of the clinical notes, we use paragraph vectors (*i.e.*, Doc2Vec) [18]. The document embeddings are passed to the convolutional layers and max-pooling layers. The output of the max-pooling layer is then flattened into Z_{note}, the latent representation of the clinical notes. Based on the type of data that we decide to consider for our experiments, the final patient-stay representation can be made of the representation of only static information (*i.e.*, Z_{static}), the

representation of only clinical notes (*i.e.*, Z_{note}), or the concatenation of the representation of clinical notes and static information (*i.e.*, $Z_{patient-stay} = [Z_{static}; Z_{note}]$). The final patient-stay representation is then passed to our classification model to detect analogies.

Classification Model. As in Alsaidi et al. [6], we adapt the neural architecture in Lim et al. [3] to our patient-stay setting. Our classification model determines if an analogy $A : B :: C : D$ is valid by verifying if A and B differ in the same way as C and D. The architecture of the classification model is a Convolutional Neural Network (CNN), which takes as input the embeddings of size n of four elements A, B, C, D. We stack them to get a matrix of size $n \times 4$. The CNN is made of three layers as depicted in the right frame of Figure 1. The first convolutional layer with 128 filters of 1 by 2 is applied on the embeddings, such that it analyses each pair separately without overlaps and measures how A and B, and how C and D differ for each component. The second convolutional layer with 64 filters of 2 by 2 is applied on the resulting matrix, after which the result is flattened into a $64 \times (n-1)$ unidimensional vector and used as input of a fully connected dense layer that produces a single output. The second layer aims at checking if the difference between A and B is the same as the one between C and D. If A and B are different in the same way as C and D, then $A : B :: C : D$ is a valid analogy. The last layer aggregates this information using a sigmoid activation to get a result (*i.e.*, output of the classification model) between 0 (for invalid analogies) and 1 (for valid analogies). All layers, except the last one, use Regularized Linear Unit (ReLU) as activation function.

3.2. Data Augmentation

Deep neural network approaches require large amounts of data. Therefore we took advantage of properties of analogies to produce additional proportions based on our dataset in a process called *data augmentation*. Previous works [19, 20, 21] have proposed postulates that analogies should obey. For this study, we consider the following:

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

Based on the definition of our analogical setting, we can apply all the above-mentioned postulates to generate valid analogical proportions except for central permutation, which can only be applied in the very particular case when $i_1 = i_2$. When $i_1 \neq i_2$, central permutation cannot be applied to increase our dataset as it would enable to associate stays of distinct patients, which is inconsistent with the aim of the *Identity* setting. Note that from reflexivity and central permutation we can deduce inner reflexivity. As reflexivity forces $i_1 = i_2$, applying it in cases where $i_1 \neq i_2$ would result in a case where $i_1 = i_2$.

For the cases where $i_1 \neq i_2$, given a valid analogy we can generate eight additional valid analogical proportions, namely

- $C : D :: A : B$,

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

and two invalid analogical proportions, namely

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

For cases where $i_1 = i_2$, we apply reflexivity to generate one more valid analogical proportion, namely $A : B :: A : B$. Note that for cases where $i_1 = i_2$, invalid analogical proportions would be considered valid.

3.3. Training and Evaluation

As mentioned, we define a *valid analogy* as a quadruple of four stays $(s_{t_1}^{i_1}, s_{t_2}^{i_1}, s_{t_3}^{i_2}, s_{t_4}^{i_2})$, where each pair of two stays belong to a single patient i_j. For each analogy in the dataset, we start by embedding the four stays. We augment the embeddings using the postulates that we recalled in Section 3.2. As a result, we generate 9 valid analogical proportions (*i.e.*, positive examples) and 2 invalid analogical proportions for cases where $i_1 \neq i_2$. For cases where $i_1 = i_2$, we obtain $10 + 2 = 12$ valid analogical proportions and no invalid analogical proportions. For optimization, we use the Binary Cross-Entropy (BCE) loss. To evaluate the classification model we use the same data augmentation process as for training, and we compute the accuracy and F1 score.

4. Dataset description

For our experiments, we used EHRs from the MIMIC-III [10] as a source of patient medical history data. MIMIC-III is a critical care database developed by the Massachusetts Institute of Technology (MIT)'s Laboratory for Computational Physiology and distributed by PhysioNet [22]. The database is publicly available, where it is accessible to researchers after finishing a HIPAA training course demanded by the National Institutes of Health (NIH). The database contains health-related information associated with all patients admitted to the ICU (Intensive Care Unit) of Beth Israel Deaconess Medical Center between the years 2001 and 2012. It encompasses data of more than 40,000 ICU patients with more than 60,000 ICU stays. All patients' data has been de-identifed in accordance with Health Insurance Portability and Accountability Act (HIPAA). The dataset contains various types of data such as patient demographics, vital signs, lab test results, medications, hospital length of stay, procedures, clinical notes, diagnosis codes (ICD-9), imaging reports, etc.

To build our dataset, we keep only adult patients (*i.e.*, patients aged 18 and above) with at least two admissions. As we do not define any order constraint, we obtain all the permutations

of all the stays belonging to a patient. We organize our dataset in way where each pair of stay is associated to the patient it belongs to: $\langle S_1, S_2, PATIENT\_ID \rangle$, where S_1 corresponds to $s_{t_1}^{i_1}$, S_2 corresponds to $s_{t_2}^{i_1}$, and the associated $PATIENT\_ID$ that represents i_1. We obtain a dataset made of 46,986 triples, where for each two pairs of stays we produce an analogy. For our experiments, we use all hospital stays associated with randomly selected 200 patients. We use the data augmentation process to generate positive and negative examples. For training and evaluation, we perform a random split (using a fixed random seed) in a training set of 70% of the extracted analogies, the remaining 30% serving as the test set. We end up with 939,638 analogies for training and 402,703 for testing. To maintain reasonable training and evaluation time, we randomly selected 50,000 analogies from the training set and 50,000 analogies from the testing set.

5. Experiment Setup

We now present the three experiments that we conducted in the *Identity* setting. In Section 5.1, we describe the patient-stay features that we consider and the data preprocessing that we performed for structured and unstructured data. We describe the implementation details in Section 5.2. The results of our experiments are reported in Section 5.3 and discussed further in this section. The code used for our experiments is written in Python 3.9 and PyTorch and is available in the repository https://github.com/Safa-98/patient-stay-analogy.

5.1. Stay Features and Data Preprocessing

We consider both structured (*i.e.*, demographics and admission-related information) and unstructured data (*i.e.*, clinical notes) to define our analogies. In this subsection, we describe the patient-stay features that are utilized by our model and some data preprocessing details.

Static information. In our experiments, our static information consists of demographic information and admission-related information. For demographic information, we extract patient's age, gender, marital status, ethnicity, and insurance information. We keep only adult patients (*i.e.*, patients aged 18 and above). We split the age into 5 groups $[18, 25[$, $[25, 45[$, $[45, 65[$, $[65, 89[$ and $[89, +\infty[$. For admission-related information, we include admission type as features.

Clinical notes. Nursing, Nursing/Other, Physician, and Radiology notes make up the majority of clinical notes in MIMIC-III database. For each hospital stay, we only kept notes that belong to these 4 categories. We excluded notes that have an error tag and notes that lack a hospital admission id.

5.2. Implementation Details

To build our corresponding cohorts, we performed the preprocessing described in the previous section to obtain our patient-stay features. Patients without any records of clinical notes or with notes that do not belong to the 4 categories defined above were removed. We computed the median of notes per hospital admission to determine the number of clinical notes to extract

Table 1
Accuracy and F1 score (both in %) of 3 runs of the classification model. Embeddings used are concatenation of static information and clinical notes.

Epochs	Valid	Invalid	F1
40 epochs	98.41 ± 1.56	68.22 ± 1.94	95.79 ± 0.59
20 epochs	94.89 ± 1.74	72.08 ± 1.68	94.30 ± 0.80
10 epochs	96.85 ± 1.75	70.31 ± 1.94	95.20 ± 0.71

per hospital admission. Therefore, we kept the first 12 notes, and used padding (*i.e.*, completion with zeros) for hospital admissions with less than 12 notes.

For the unsupervised Doc2Vec model [18], we finetune it on the training set to obtain the document-level embeddings using the Gensim toolkit [23]. For the training algorithm, we use PV-DBOW (Paragraph vector-Distributed Bag of Words). We set the number of training epochs as 30, the initial learning rate as 0.025, the learning rate decay as 0.0002, and the dimension of vectors as 200 to train. The Fusion CNN model is trained with Adam optimizer with a learning rate of 0.0001 and ReLU as the activation function. The chosen batch size is 64.

In this paper we perform three experiments. In the first, we consider both structured and unstructured data. Therefore, we obtain our patient-stay representation by concatenating the representations of clinical notes along with static information. In this experiment, we verify if a particular hospital stay belongs to a patient by looking at both the structured and unstructured data associated with each stay. In the second, we only consider unstructured data, which means that our patient-stay representations are based solely on the representations of clinical notes. Therefore, by looking at clinical notes associated with a single hospital stay, we check if a particular hospital stay belongs to a patient. In the third, we only consider structured data, which means that our patient-stay representations are based solely on the representations of static information (*i.e.*, demographics and admission-related information). Therefore, we verify if a particular hospital stay belongs to a patient by looking at the static information that is associated with a hospital stay.

5.3. Results and Discussion

As mentioned previously, we conducted three experiments that mainly differ in what type of data was used to obtain our patient-stay representations. For all the experiments, we used 50,000 analogies for training and evaluation, and applied the same procedure for data augmentation. We report the accuracy and F1 score for each experiment. The F1 score gives a better measure of the incorrectly classified cases than the accuracy metric.

For the first experiment, we fed our embedding model with both structured (*i.e.*, demographics and admission-related information) and unstructured data (*i.e.*, clinical notes). Our patient-stay representations are thus made of the concatenation of static information and clinical notes. We chose the epochs where the training loss is at the local minimum. We trained our model for 10, 20, and 40 epochs, with 3 different random initializations in each case. Our results are detailed in Table 1. Our model performs the best for positive examples. For 40 epochs, the model gives the best result for valid analogies and performs best for invalid analogies for 20 epochs.

Table 2
Accuracy and F1 score (both in %) of 3 runs of the classification model. Embeddings used are based on only clinical notes.

Epochs	Valid	Invalid	F1
40 epochs	88.52 ± 6.91	77.20 ± 6.92	95.64 ± 1.27
20 epochs	97.71 ± 1.69	67.89 ± 2.43	94.74 ± 0.93
15 epochs	92.05 ± 6.42	74.45 ± 7.39	94.90 ± 1.25

Table 3
Accuracy and F1 score (both in %) of 3 runs of the classification model. Embeddings used are based on only static information.

Epochs	Valid	Invalid	F1
40 epochs	99.98 ± 0.001	66.14 ± 0.04	96.37 ± 0.01
20 epochs	99.98 ± 0.004	65.96 ± 0.31	96.35 ± 0.03
15 epochs	99.98 ± 0.002	66.12 ± 0.13	96.33 ± 0.05

For our second experiment, we used only unstructured data, *i.e.*, the Z_{note} part of the embedding for the patient-stay representations. Our patient-stay representations thus consisted of only the representation of clinical notes. The training loss was at the local minimum for 15, 20, and 40 epochs. Therefore, we trained our model for 15, 20, and 40 epochs, with 3 different random initializations in each case. As shown in Table 2, our model performs the best for positive examples when we train by 20 epochs.

For our third experiment, we used only structured data, *i.e.*, Z_{static}, to represent our patient-stay representations. Our training loss was at the local minimum for 15, 20, and 40 epochs. Therefore, we trained our model for 15, 20, and 40 epochs, with 3 different random initializations in each case. We report our results in Table 3. As seen, the accuracy for positive examples is high for all cases compared to negative examples where the accuracy drops.

In all our experiments, we can see that our model performs the best for positive examples regardless of whether we use $[Z_{static}; Z_{note}]$, only Z_{note}, or only Z_{static} for the patient-stay representations. This can be explained as a result of the imbalance between positive and negative examples in the training data. Balancing the data would be the next step as it proved to be a good solution for [6] to get similar results for positive and negative examples. The accuracy for valid analogies is the highest when our embedding model is fed with only static information. Between the first and the second experiment, the accuracy is the highest for valid analogies when the patient-stay representations are made of the concatenation in contrast to when our patient-stay representations are made of only clinical notes. This indicates that adding or using static information when learning patient-stay representations, as in the first and third experiment, improves the performance of our model, where it allows the model to better distinguish the stays and to match them to the patient they belong to. We also notice that the accuracy for invalid analogies is the highest when the embedding model is fed with only clinical notes. For all performed experiments, the F1 score is high, which indicates that our model is able to correctly classify analogies to the class they belong to (*i.e.*, valid or invalid).

To gain more insight into how our models perform, we conducted an error analysis where we noticed that most misclassifications were spotted in two cases.

1. **Cases where $i_1 = i_2$.**
   To recall, we do not generate invalid analogies for cases where $i_1 = i_2$; therefore, invalid analogy forms ($D : A :: B : C$ and $A : C :: B : D$) should be considered valid in these cases. In our error analysis, we noticed that when the four stays belong to the same patient, our model classifies the above-mentioned invalid analogy forms as invalid instead of valid. We believe that our model was not trained enough to distinguish these forms of analogies as there were less analogies with four stays belonging to the same patient generated in our dataset.
2. **Cases where representations are made of only clinical notes.**
   To recall, in our second experiment we only used the representations of clinical notes to obtain patient-stay representations. We noticed that when the category of the clinical notes is similar between two hospital stays or when two hospital stays have less than five clinical notes, our model struggles to distinguish between the two hospital stays. This indicates that in some cases using only clinical notes to learn patient-stay representations might not be sufficient as these notes might not contain enough information to help our model differentiate between two similar stays that belong to two distinct patients. As a result, the model would incorrectly match these two similar stays to the same patient.

In these experiments, we did not include temporal data, where we only used demographics and admission-related information as structured data. It would be interesting to also include temporal signals (*i.e.*, vital signs) along with demographics and admission-related information as structured data. Our patient-stay representations would be then made of the concatenation of the representations of static information and temporal signals as structured data and the representation of clinical notes as unstructured data.

6. Conclusion and Perspectives

We adapted the approach in [3, 6] from semantic and morphological analogies to patient-stay analogies. Our prototypical architecture has some limits, but seems promising for the task of patient identification. Our classification model is flexible in terms of the analogies that it classifies. Changing the way the data is augmented will change the way the model behaves. Our model can be adapted to different healthcare applications through dedicated embedding models [24]. Inspired by [14], we implemented a model to build patient-stay representations. As mentioned in Section 5.3, there are multiple plausible improvements to our approach, in terms of balancing valid and invalid analogies as well as including other types of data to build our patient-stay representations. As we limited ourselves to analogy detection, a future work would be to address analogy solving in the same setting that would allow the generation of synthetic patient-stays.

Acknowledgments

Experiments presented in this paper were carried out using computational clusters equipped with GPU from the Grid'5000 testbed (see https://www.grid5000.fr).

The research work of the second and third named authors is partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation program under GA No 952215, and the Inria Project Lab "Hybrid Approaches for Interpretable AI" (HyAIAI).

References

[1] R. Fam, Y. Lepage, Morphological predictability of unseen words using computational analogy, in: Workshops Proceedings for the Twenty-fourth International Conference on Case-Based Reasoning (ICCBR), volume 1815, 2016, pp. 51–60.

[2] S. E. Reed, Y. Zhang, Y. Zhang, H. Lee, Deep visual analogy-making, in: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2015, pp. 1252–1260.

[3] S. Lim, H. Prade, G. Richard, Solving word analogies: A machine learning perspective, in: Proceedings of the Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU), volume 11726, 2019, pp. 238–250.

[4] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[5] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, D. Salesin, Image analogies, in: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), 2001, pp. 327–340.

[6] S. Alsaidi, A. Decker, P. Lay, E. Marquer, P.-A. Murena, M. Couceiro, A neural approach for detecting morphological analogies, in: Proceedings of the 8th IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2021, pp. 1–10.

[7] M. A. Casteleiro, J. D. Diz, N. Maroto, M. J. F. Prieto, S. Peters, C. Wroe, C. S. Torrado, D. M. Fernandez, R. Stevens, Semantic deep learning: Prior knowledge and a type of four-term embedding analogy to acquire treatments for well-known diseases, JMIR Medical Informatics 8 (2020) 1–28.

[8] E. Dynomant, R. Lelong, B. Dahamna, C. Massonnaud, G. Kerdelhué, J. Grosjean, S. Canu, S. J. Darmoni, et al., Word embedding for the french natural language in health care: comparative study, JMIR medical informatics 7 (2019) 118–122.

[9] N. N. Rather, C. Patel, S. A. Khan, Using deep learning towards biomedical knowledge discovery, International Journal of Mathematical Sciences and Computing, (IJMSC) 3 (2017) 1–10.

[10] A. E. W. Johnson, T. J. Pollard, L. Shen, L. wei H. Lehman, M. Feng, M. M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, R. G. Mark, Mimic-iii, a freely accessible critical care database, Scientific Data 3 (2016).

[11] Y. Si, J. Du, Z. Li, X. Jiang, T. Miller, F. Wang, W. Jim Zheng, K. Roberts, Deep representation learning of patient data from electronic health records (ehr): A systematic review, Journal of Biomedical Informatics 115 (2021) 1–42.

[12] J. Zhang, K. Kowsari, J. H. Harrison, J. M. Lobo, L. E. Barnes, Patient2vec: A personalized interpretable deep representation of the longitudinal electronic health record, IEEE Access 6 (2018) 65333–65346.

[13] S. Madhumita, S. Simon, L. Kim, D. Walter, Patient representation learning and interpretable evaluation using clinical notes, Journal of biomedical informatics 84 (2018) 103–113.

[14] D. Zhang, C. Yin, J. Zeng, X. Yuan, P. Zhang, Combining structured and unstructured data for predictive models: a deep learning approach, BMC Medical Informatics and Decision Making 20 (2020) 280.

[15] P. Waruhari, A. Babic, L. Nderu, M. C. Were, A review of current patient matching techniques, in: Informatics Empowers Healthcare Transformation (ICIMTH), volume 238, 2017, pp. 205–208.

[16] F. N. Wirth, T. Meurers, M. Johns, F. Prasser, Privacy-preserving data sharing infrastructures for medical research: systematization and comparison, BMC Medical Informatics Decision Making 21 (2021) 242.

[17] B. H. Just, D. T. Marc, M. Munns, R. H. Sandefer, Why patient matching is a challenge: Research on master patient index (mpi) data discrepancies in key identifying fields., Perspectives in health information management 13 (2016) 1e.

[18] Q. V. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31th International Conference on Machine Learning (ICML), volume 32, 2014, pp. 1188–1196.

[19] L. Miclet, S. Bayoudh, A. Delhay, Analogical dissimilarity: Definition, algorithms and two experiments in machine learning, Journal of Artificial Intelligence Research 32 (2008) 793–824.

[20] Y. Lepage, De l'analogie rendant compte de la commutation en linguistique, Habilitation à diriger des recherches, Universit'e Joseph-Fourier - Grenoble I, 2003.

[21] C. Antic, Analogical proportions, ArXiv abs/2006.02854 (2020).

[22] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley, Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals., Circulation 101 23 (2000) E215–20.

[23] R. Řehůřek, P. Sojka, Software Framework for Topic Modelling with Large Corpora, in: Proceedings of the LREC Workshop on New Challenges for NLP Frameworks, 2010, pp. 45–50.

[24] S. Alsaidi, M. Couceiro, A. Burgun, N. Garcelon, A. Coulet, Exploring analogical inference in healthcare, in: Workshop on Interactions between Analogical Reasoning and Machine Learning (IARML), 2022 (to appear).

The Application of Qualitative Metadata to Analogical Reasoning

Dave Raggett

W3C/ERCIM, Sophia Antipolis, France

Abstract

Analogical reasoning can be used for plausible inferences based upon direct similarities or structural mappings involving properties and relationships. This can be implemented on top of a combination of symbolic knowledge plus sub-symbolic qualitative metadata, with matching based upon structural or causal similarities, and noticing interesting differences, in essence, abstracting from similarities and dissimilarities, and will be applied to examples of the form "A is to B as C is to ?X". A further challenge is to support the use of literal and figurative analogies in natural language, e.g., comparing life to the wheel of fortune, when you want to highlight the role of chance. An easy-to-use syntax will be presented for expressing knowledge, along with a web-based proof of concept demonstrator, and a unifying cognitive architecture for human-like AI. This builds upon pioneering work by Alan Colins on plausible reasoning, and Dedre Gentner on analogies.

Keywords

Plausible reasoning, Human-like AI, analogies, Cognitive Architecture

1. Introduction

The paper starts with an introduction to plausible reasoning before moving on to analogical reasoning and how this can be supported as an extension of plausible reasoning. This very much work in progress, and part of a long term drive to realise human-like memory, reasoning and learning in cognitive agents.

2. Plausible Reasoning

We are learning all the time, and revising our beliefs and understanding as we interact with others. As such our knowledge is imperfect and subject to uncertainties, incompleteness and inconsistencies. This is challenging both for conventional mathematical logic, and for statistical approaches such as Bayesian inference due to the difficulties in obtaining the required statistics. Evolution has equipped humans with the means to deal with imperfect knowledge in a rational way based upon sound judgement, albeit subject to various kinds of cognitive biases, see, e.g., Daniel Kahneman [1].

ICCBR Analogies'22: Workshop on Analogies: from Theory to Applications at ICCBR-2022, September, 2022, Nancy, France

dsr@w3.org (D. Raggett)

People have studied the principles for plausible arguments since the days of Ancient Greece, e.g., Carneades and his guidelines for argumentation. This was developed further by a long line of philosophers, including Locke, Bentham, Wigmore, Keynes, Wittgenstein, Pollock and many others. Plausible reasoning is everyday reasoning, and the basis for legal, ethical and business discussions. It is now timely to exploit plausible reasoning with imperfect knowledge in support of human-machine cooperative work. This will enable computers to analyse, explain, justify, expand-upon and argue in human-like ways.

Consider $A \rightarrow B$, which means if A is true then B is true. If A is false then B may be true or false. If B is true, we still can't be sure that A is true, but if B is false then A must be false. We can go further with a little knowledge. Consider a more concrete example: if it is raining then it is cloudy. This can be used for inferences in both directions. Rain is more likely if it is cloudy, and likewise, if it is not raining, then it might be sunny, so it is less likely that it is cloudy. Such arguments draw upon qualitative terms in lieu of quantitative statistics.

In essence, plausible reasoning draws upon prior knowledge as well as on the role of analogies, and the consideration of examples as precedents. Mathematical proof is replaced by reasonable arguments, both for and against a premise, along with how these are assessed. In legal proceedings, for instance, arguments are laid out by the Prosecution and the Defence, the Judge decides what evidence is admissible, whilst guilt is assessed by the Jury.

During the 1980's Alan Collins and his co-workers developed a theory of plausible reasoning [2] based upon analysis of recordings of how people reasoned. They found that:

- There are several categories of inference rules that people commonly use to answer questions.
- People weigh the evidence that bears on a question, both for and against, rather like in court cases.
- People are more or less certain depending on the certainty of the premises, the certainty of the inferences, and whether different inferences lead to the same or oppositive conclusions.
- Facing a question for which there is an absence of directly applicable knowledge, people search for other knowledge that could help given potential inferences.

Plausible knowledge can be expressed using a combination of symbolic graphs and associated metadata. This paper introduces the plausible knowledge notation (PKN) as an easy-to-read extensible syntax accompanied with an implementation as a JavaScript library for use in web page demos for different kinds of plausible reasoning [3], and as part of work for the W3C Cognitive AI Community Group [4]. PKN supports a variety of different kinds of statements:

Properties

```
flowers of England includes daffodils, roses (certainty high)
```

where `flowers` is a property of the referent `England`, and the use of `includes` signifies that the property is an open set with values `daffodils` and `roses`. For a closed set, use `is` instead of `includes`. Trailing round brackets are used to list qualitative metadata, in this case declaring that the statement has a high certainty.

Relationships

```
robin kind-of songbird
duck similar-to goose for habitat
duck dissimilar-to goose for neck-length
```

where robin is declared as a subclass of songbird, and duck is declared as being similar to goose for habitat and dissimilar to goose in respect to neck length.

Dependencies

```
climate depends-on latitude
pressure decreases-with altitude
current increases-with voltage
```

where climate depends on latitude in some unspecified way, whilst pressure decreases with increasing latitude and current increases with increasing voltage.

Implications

```
temperature of ?place is warm &
rainfall of ?place is heavy
implies grain of ?place includes rice
```

Implications are a form of if-then rules where variables are prefixed with a question mark.

Metadata can be given with all kinds of PKN statements. Relationships, dependencies and implications can be used for inferences in both directions, subject to any associated metadata. Following Collins, PKN supports several kinds of statement metadata relevant to different kinds of inferences:

Typicality in respect to other group members, e.g., robins are typical song birds.

Similarity to peers, e.g., having a similar climate.

Strength as conditional likelihood, e.g., the strength of climate for determining which kinds of plants grow well. The forward and backward strengths may differ, e.g., rain is a strong indicator of cloudy weather, whilst cloudy weather is a weak indicator of rain.

Frequency as the proportion of children with a given property, e.g., most species of birds have the ability to fly.

Dominance as the relative importance in a given group, e.g., the size of a country's economy.

Multiplicity as the number of items in a given range, e.g., how many different kinds of flowers grow in England.

The web demonstrator [3] allows you to pick from an assortment of queries, and to then see a trace of the reasoning, proceeding from the facts to the premise. The inference engine itself works backwards from the premise to the facts, and the explanation is subsequently generated from the trace of execution. Here is an example of the reasoning associated with the query whether daffodils are grown in England:

```
Premise: flowers of England includes daffodils
Evidence supporting the premise:

  flowers of England includes temperate-flowers
  and daffodils kind-of temperate-flowers
  therefore flowers of England includes daffodils

  flowers of Netherlands includes daffodils, tulips
  and Netherlands similar-to England for flowers
  therefore flowers of England includes daffodils

Suggesting: flowers of England includes daffodils is likely
```

This develops two lines of argument in favour of the premise in the query. The first is based on recognising that daffodils are a sub-class of temperate flowers, which are known to grow in England. The second makes use of knowledge that England and the Netherlands are similar in respect to the flowers grown. The inference engine uses a fixed strategy for searching for and applying relevant inferences. This may involve the use of graph algorithms such as spreading activation to propose and prioritise potential inferences as suggested by Collins. Other algorithms are used to compute certainties of inferences based upon statement metadata, and for assessing and combining multiple lines of argument. Future work will explore a wide range of reasoning, including spatial, temporal, causal and social reasoning, along with metacognition for problem solving, and support for System 1 and 2 cognition [1].

3. Analogical Reasoning

What benefits are potentially possible for analogical reasoning by cognitive agents? A starting point is to distinguish between literal and figurative analogies. The former involves things that are really quite similar, whilst the latter are not obviously comparable at first glance. Analogies can help agents to generalise their knowledge based upon a few examples. This has potential applicability for the properties of things, understanding their behaviours, as well as for problem solving by drawing upon previous experience in similar situations.

Analogies are further related to similes and metaphors in language. Similes involve a comparison that explicitly emphasises some comparable characteristic, e.g., "his words were like a punch in the guts" as a way to establish the impact of the words on the listener, whilst metaphors involve an implicit comparison, e.g., "to get cold feet" is to have second thoughts about some proposed course of action. People commonly use similes and metaphors to communicate thoughts in ways that are more vivid and interesting, as well as to structure perceptions and understanding, see Lackoff and Johnson [5]. As such, this is expected to be an important aspect of human-machine communication, albeit one that is very challenging, at least in the near future.

Dedre Gentner [6] notes that analogies may involve matching based upon structural or causal similarities, and noticing interesting differences, in essence, abstracting from similarities and dissimilarities

Gentner cites the example of plumbing in that electrical circuits can be likened to a plumbing system for water, e.g., equating voltage to pressure, and electrical current to water flow. Causal relationships for the source can be used to suggest similar relationships for the target, e.g., higher voltage leads to greater current just as higher water pressure leads to greater water flow.

Two situations can be identified as similar if they share some of the same properties, with the implication that you may be able to infer properties of the target from properties of the source. You may also be able to infer relationships, e.g., part/whole or cause/effect. More generally, the situations have different properties, that can however be mapped one to another (as in voltage to pressure). Such mappings have to be learned or guessed from matching relationships. Thus, if two situations/contexts have several properties or relationships in common, then we may consider them as analogical equivalents.

The notion of similarity introduced by Collins [2] supports inferences on shared property values at least in some given context, see the `similar-to` and `dissimilar-to` statements in PKN above. A generalisation is to relate pairs of different properties, e.g., voltage corresponds to pressure, and current to flow when making an analogy between electrical circuits and plumbing. Such pairings can be represented by adding a `corresponds-to` statement to PKN:

```
voltage corresponds-to pressure for circuit
current corresponds-to flow for circuit
flow increases-with pressure
# thus allowing us to infer
current increases-with voltage
```

We also need a way to describe that voltage and current are characteristics of electrical circuits, which are a sub-class of circuits, e.g.

```
electrical-circuit kind-of circuit
voltage property-of electrical-circuit
current property-of electrical-circuit
```

An open question is how people learn such knowledge from examples and being taught by others. That relates to the notion of syntagmatic and paradigmatic learning. Syntagmatic learning deals with learning co-occurrence patterns within episodes, whilst, paradigmatic learning involves identifying generalisations, and is believed to develop at a later age in childhood.

Analogies as part of critical thinking. It is easy to find web sites that propose the use of analogies for teaching purposes. These are based upon simple patterns, e.g., synonyms, antonyms, part/whole, cause/effect, etc. Here are some examples:

```
battery is-to torch as ?x is-to car # engine powers a car
itch is-to scratch as ?x is-to cold # virus causes a cold
wall is-to brick as bottle is-to ?x # a bottle is made of glass
```

Solving such queries involves identifying the pattern, and then applying background knowledge. The first step is to recognise the query as using an analogy. The next step is to use the pair that doesn't involve a variable to identify likely patterns, e.g., battery/torch in the first example. The knowledge base may contain plenty of facts and relationships, and it will be important

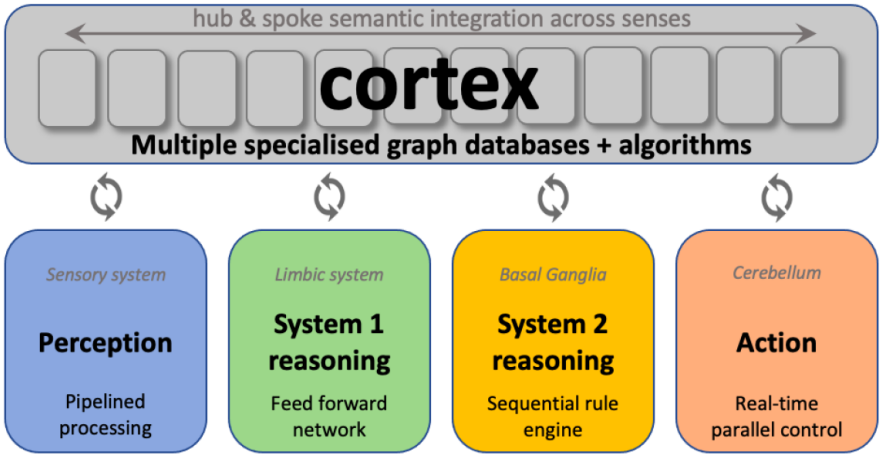

Figure 1: Cognitive architecture with multiple cognitive circuits loosely equivalent to shared blackboards – where semantic integration mimics the role of the Anterior Temporal Lobe as a hub for unimodal spokes.

to look for patterns that also occur for the pair with the variable. It may be the case that two pairs use different relationships, in which case, we need to find plausible evidence that they are comparable patterns.

Simple analogies are amenable to a fixed strategy plus associated graph algorithms. Qualitative metadata can be used to reason about certainty and to prioritise processing. What about more complicated analogies? The work by Jaime Carbonell on derivational analogies [7] is inspiring. The paper describes a problem solver that searches for analogies with previously solved problems, adapting the solution as needed based upon an analysis comparing the old and new problems.

4. Cognitive Architecture

The quest for realising human-like AI owes a huge debt to many pioneers over many decades. To mention just a few: Daniel Kahneman, a Nobel-prize willing psychologist who studied System 1 & 2 thinking along with cognitive biases [1]; Philip Johnson-Laird, a cognitive scientist renowned for his work on how humans reason in terms of mental models rather than logic and statistics [8]; John R. Anderson, a cognitive scientist renowned for his work on the ACT-R cognitive architecture for sequential cognition [9]; and Alan Collins, a cognitive scientist renowned for his work on plausible reasoning and intelligent tutoring systems [2].

Figure 1 illustrates a high-level cognitive architecture inspired by the structure and function of the human brain.

Memory is based on graph databases and associated graph algorithms. It combines symbolic graphs with sub-symbolic information, mimicking the human cortex, and defined at a conceptual level above that of RDF and Property Graphs (including NGSI-LD). Recall is stochastic reflecting prior knowledge and past experience. This involves activation boost/decay, spreading activation, the forgetting curve and spacing effect.

Perception interprets sensory data at progressively higher levels of abstraction, and places the resulting models into the cortex. Cognitive rules can set the context for perception, and direct attention as needed. Events are signalled by queuing chunks to cognitive buffers to trigger rules describing the appropriate behaviour. A prioritised first-in first-out queue is used to avoid missing closely spaced events.

System 1 is about intuitive/emotional thought, and prioritising what's important. The limbic system provides rapid automatic assessment of past, present and imagined situations without the delays incurred in deliberative thought. Emotions are perceived as positive or negative, and associated with passive or active responses, involving actual and perceived threats, goal-directed drives and soothing/nurturing behaviours.

System 2 is slower and more deliberate thought, involving sequential execution of rules to carry out particular tasks, including the means to invoke graph algorithms in the cortex, and to invoke operations involving other cognitive circuits. Thought can be expressed at many different levels of abstraction, and is subject to control through metacognition, emotional drives, internal and external threats.

Action is about carrying out actions initiated under conscious control, leaving the mind free to work on other things. An example is playing a musical instrument where muscle memory is needed to control your finger placements as thinking explicitly about each finger would be far too slow. The cerebellum provides real-time coordination of muscle activation actively guided by perception.

This architecture has been partially realised with a suite of web-based demos developed for the W3C Cognitive AI Community Group [4]. This includes the chunks and rules specification, and an implementation as a JavaScript library. Chunks are essentially collections of name/value pairs, where values are literals or references to other chunks, or lists thereof. Chunks are associated with decaying activation levels to mimic the characteristics of human-memory. Chunk rules support sequential reasoning (System 2).

Further work is underway to incrementally realise the requirements identified by Kahneman for System 1, and to understand how plausible reasoning, learning and metacognition can be layered on top of System 1 and 2. This will include the intuitive and deliberative reasoning involved in natural language processing, and the human ability to reason about the past, present, and imagined situations.

5. Conclusions

AI today can be broadly split into symbolic AI, statistical techniques, and approaches based upon deep learning and multi-layer artificial neural networks. Work in the cognitive sciences suggests a middle ground that combines symbols and sub-symbolic metadata, and is open to distributed representations (e.g., as vectors in noisy high dimensional spaces) where this would yield computational benefits. Traditional symbolic AI is hard to scale, relying on hand-coded knowledge, along with difficulties in dealing with imperfect knowledge, whilst deep learning scales well, but has challenges with reasoning and transparency. This paper draws attention

to the potential for mimicking human-like memory, reasoning and learning, inspired by the wealth of research in the cognitive sciences.

References

[1] D. Kahneman, Thinking, fast and slow, Macmillan, 2011.

[2] A. M. Collins, R. S. Michalski, The logic of plausible reasoning: A core theory, Cogn. Sci. 13 (1989) 1–49. URL: https://doi.org/10.1207/s15516709cog1301_1. doi:10.1207/s15516709cog1301_1.

[3] D. Raggett, Plausible reasoning demo, visited on July 24, 2022. URL: https://www.w3.org/Data/demos/chunks/reasoning/.

[4] W3C Cognitive AI Community Group, visited on July 24, 2022. URL: https://github.com/w3c/cogai#readme.

[5] G. Lakoff, M. Johnson, Metaphors we live by, University of Chicago press, 1980.

[6] D. Gentner, Structure-mapping: A theoretical framework for analogy, Cogn. Sci. 7 (1983) 155–170. URL: https://doi.org/10.1207/s15516709cog0702_3. doi:10.1207/s15516709cog0702_3.

[7] J. G. Carbonell, M. Veloso, Integrating derivational analogy into a general problem solving architecture, in: Proceedings of the First Workshop on Case-Based Reasoning, Morgan Kaufmann Tampa, FL, 1988, pp. 104–124.

[8] P. Johnson-Laird, How we reason, Oxford University Press, 2006.

[9] J. R. Anderson, How can the human mind occur in the physical universe?, Oxford University Press, 2007.

Towards efficient scoring of student-generated long-form analogies in STEM

Thilini Wijesiriwardene[1,*], Ruwan Wickramarachchi[1], Valerie L. Shalin[1,2] and Amit P. Sheth[1]

[1]AI Institute, University of South Carolina, Columbia, SC, USA
[2]Department of Psychology, Wright State University, Dayton, OH, USA

Abstract

Switching from an analogy pedagogy based on comprehension to analogy pedagogy based on production raises an impractical manual analogy scoring problem. Conventional symbol-matching approaches to computational analogy evaluation focus on positive cases, and challenge computational feasibility. This work presents the Discriminative Analogy Features (DAF) pipeline to identify the discriminative features of strong and weak *long-form* text analogies. We introduce four feature categories (semantic, syntactic, sentiment, and statistical) used with supervised vector-based learning methods to discriminate between strong and weak analogies. Using a modestly sized vector of engineered features with SVM attains a 0.67 macro F1 score. While a semantic feature is the most discriminative, out of the top 15 discriminative features, most are syntactic. Combining this engineered features with an ELMo-generated embedding still improves classification relative to an embedding alone. While an unsupervised K-Means clustering-based approach falls short, similar hints of improvement appear when inputs include the engineered features used in supervised learning.

Keywords

Descriptive analogies, Analogical features, Analogy scoring, Long-form analogies,

1. Introduction

Analogical reasoning relies on the ability to draw on the relational similarities between two systems of objects in different contexts [1, 2, 3]. Analogies appear in several disciplines such as engineering design, scientific reasoning, and often in STEM education. However, the dominant pedagogical paradigm requires students to comprehend curated analogies. In this work, we are focusing on the evaluation of *student-generated* analogies in their first undergraduate biochemistry course.

Problem sets, specifically created to explore the underlying mechanisms of analogical reasoning, consist of visual and verbal analogies [4, 5]. Verbal analogies have two primary forms; analogical proportions and long-form analogies. Analogical proportions follow a four-term

ICCBR Analogies'22: Workshop on Analogies: from Theory to Applications at ICCBR-2022, September, 2022, Nancy, France

*Corresponding author.
thilini@sc.edu (T. Wijesiriwardene); ruwan@email.sc.edu (R. Wickramarachchi); valerie.shalin@wright.edu (V. L. Shalin); amit@sc.edu (A. P. Sheth)
0000-0001-8431-8443 (T. Wijesiriwardene); 0000-0001-5810-1849 (R. Wickramarachchi); 0000-0001-8135-2793 (V. L. Shalin); 0000-0002-0021-5293 (A. P. Sheth)

format such as "*A* to *B* as to *C* to *D*" or *A*:*B*::*C*:*D* [6]. Recent work on computational analogy making focuses on analogical proportions [7, 8, 9]. Our interest here lies in *long-form* analogies consisting of a narrative/ description of a target unfamiliar situation/ system (for the context or lesson to be learned) using several sentences and a familiar source (base) [10, 3]. While the objects across the two descriptions differ, they employ similar *relations* between these objects. An example of a well-known long-form analogy is between the solar system (source) and the Rutherford-Bohr model of the atom (target) [11] where small objects revolving around a large central object provide relational similarity with the target. The solar system and the atom can each be described using several sentences. Parallels between these two systems can then be drawn, making the two descriptions analogous.

The atom-solar system analogy exemplifies the curated analogies in STEM textbooks. [12] has developed algorithms for evaluating correct or slightly incorrect long-form analogies. In [13] we solicited analogies from STEM students, with the expectation, relative to a comprehension exercise, that analogy production is both more engaging and allows students to employ existing familiar knowledge to scaffold the acquisition of new knowledge. No matter how pedagogically successful, manual scoring is impractical for an analogy production pedagogy. Production pedagogy elevates an analogy scoring problem for computational solution.

We aim to identify discriminative features between strong and weak, long-form *student generated* verbal analogies collected in a college biochemistry class (see Section 2.1) to support efficient computational scoring. To this end, we develop the Discriminative Analogy Features (DAF) pipeline.

We use a long-form analogy dataset, instructor-graded as strong or weak. We explore both supervised and unsupervised learning classifiers using vectors based on embeddings, engineered features and both. Given manually annotated data for a supervised learning classifier (i.e. SVM), we identify the discriminative features of strong and weak analogies.

We introduce DAF, a pipeline to identify the discriminative features of strong and weak analogies. We also introduce four feature categories – semantic, syntactic, sentiment, and statistical used in supervised learning to discriminate between strong and weak analogies. We show that "unique attribute count", a *semantic feature*, is the most discriminative when identifying between strong and weak analogies. Out of the top 15 discriminative features, most are *syntactic*. Unsupervised learning is unable to obtain comparable success, though it slightly improves with features corresponding to the above categories.

The rest of this paper is organized as follows: Section 2 introduces and describes the DAF pipeline and identifies the discriminative features. Section 3 presents the discussion with findings, insights, limitations, and future work subsections. Section 4 concludes the paper.

2. Discriminative Analogy Features (DAF) Pipeline

To identify discriminative features, we introduce the pipeline illustrated in Figure 1. In the subsequent subsections, we describe each pipeline component.

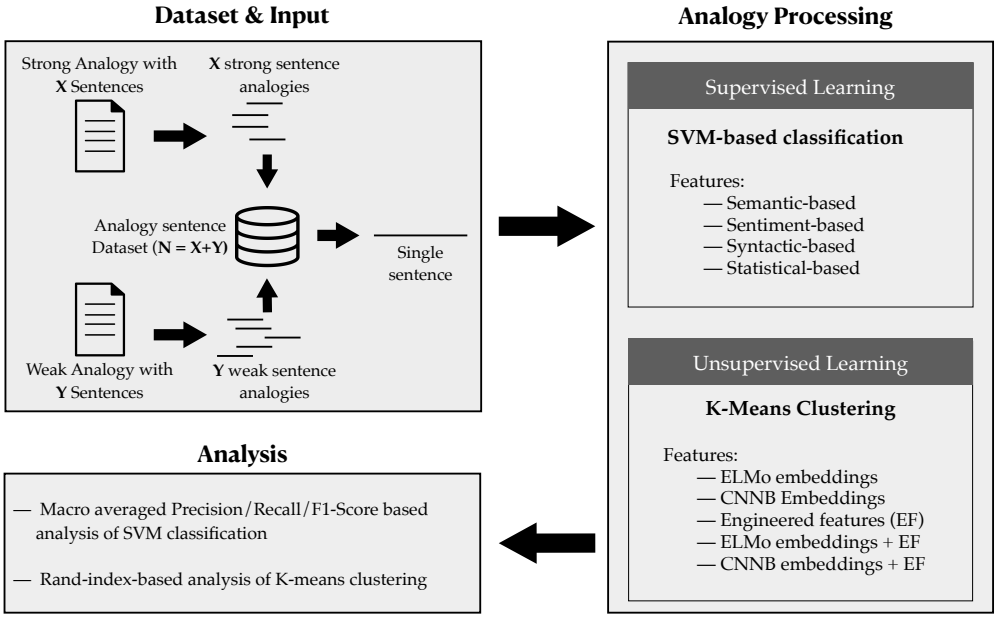

Figure 1: Illustration of the DAF Pipeline. The analogies are split into sentences to create the analogy dataset. Single sentences are sent through Analogy Processing. Features of the sentences are extracted and used in SVM-based classification and K-means clustering. An analysis is then conducted on SVM-based classification and K-means clustering results.

2.1. Dataset and Input

The dataset used in this work was drawn from 500 student-created analogies, collected in a college classroom. An instructor explained a process in the domain of biochemistry, e.g., Glycolysis (source analogy), and requested students to construct a scenario analogous to the explained process from a domain of their choice (target analogy). The instructor then evaluated 31 student-generated analogous scenarios as a strong or weak analogy based on its correspondence to the Biochemistry concept. A strong analogy corresponds well with the target analogy, and a weak analogy minimally corresponds with the target analogy. To increase the size of the 31 exemplar data set from the original we split each analogy into its constituent sentences, generating a data set of 526 strong exemplars and 140 weak exemplars. Each constituent sentence of an analogy falls into the same annotation category as the original analogy. *Ergo, the initial input to the DAF pipeline is a sentence.* This work does not distinguish between the analogy's target domains (Enzyme Kinetics and Glycolysis). Table 1 presents the summarized statistics of the dataset.

Table 1
Dataset statistics

	Strong	Weak
Num. of analogies	25	6
Num. of analogies (sentences)	586	140

2.2. Input Processing

Sentences were processed and used as inputs to a Support Vector Machine (SVM) classifier (supervised learning) and K-means clustering (unsupervised learning) separately. In the following section we briefly review the background of input processing techniques, learning methods and implementation details.

2.2.1. Background

SVM is a supervised learning technique that creates functions to map inputs to pre-existing annotations [14]. SVM is an easy-to-interpret classifier providing competitive performance in classification, regression, and outlier detection tasks [15]. The following paragraphs detail the background of four feature groups of interest here.

The obviously relevant features are semantic. Abstract Meaning Representation (AMR) is a semantic representation language that expresses a sentence's logical meaning by converting it to a rooted, directed, acyclic, edge-labeled, and leaf-labeled graph. [16]. To abstract away from syntactic idiosyncrasies, AMR assigns the same AMR graph to sentences with the same *meaning*. Nodes of an AMR graph are labeled as *concepts*, edges as *relations*, and concept properties as attributes. Concepts are either English words, PropBank framesets [17] or special keywords. There are approximately 100 relations [16]. AMR is used as a semantic representation of text in several NLP tasks such as summarization [18], machine comprehension [19], and event extraction [20, 21]. In this work we use AMR representations to extract concepts, relations and attributes present in sentence analogies. Figure 2 illustrates the AMR for a sentence from the dataset.

Sentiment-based features potentially reveal student engagement. Sentiment analysis aims to identify emotional or affective tendencies in user-generated content such as tweets, product reviews, and feedback [22]. Subjectivity detection and polarity determination are two common tasks in sentiment analysis [23]. Subjectivity quantifies the personal opinions versus factual information contained in the text. High subjectivity indicates the text contains more personal opinions compared to factual information [23]. Polarity describes the sentiment of a piece of text as positive, negative, or neutral [22].

We extract three groups of syntactic features. The first feature group is Part of Speech (POS), a grammatical classification of the word types in a sentence. These POS tags commonly include nouns, verbs, adjectives, etc. [24]. Named Entities Recognition (NER), the second feature group, is used to identify occurrences of named entities such as people, organizations, times, and locations in a sentence [25]. The third feature group is sentence type. Sentences in the dataset are identified as complex or compound sentences and simple sentences. In linguistics, complex sentences are sentences with two or more clauses connected with a subordinate conjunction. Simple sentences contain one independent clause [26].

We use four routine and straightforward statistical features, word count, character count, the average word length of a sentence (character count/ word count), and the number of unique words in a sentence.

K-means is a non-deterministic, iterative, and unsupervised machine learning technique to produce clusters from data [27]. Unsupervised learning here serves as both a baseline for

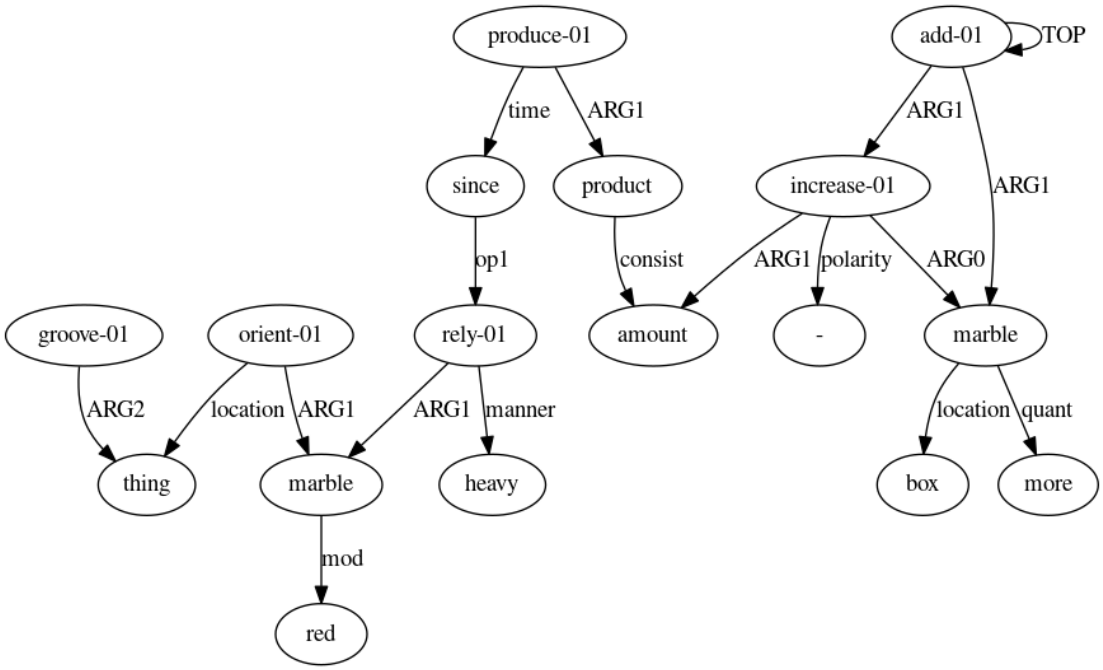

Figure 2: AMR representation of the sentence *"Adding more marbles to the box will not increase the amount of product produced since it relies heavily on red marbles being oriented into the grooves."* from the dataset.

comparison with supervised learning results, and as a long term goal in itself, independent of any manual annotation. In the simplest test, we converted the input sentences to embeddings and cluster them using K-means clustering. Sentence embeddings were created using two techniques, context-based Embeddings from Language Models (ELMo) and knowledge-graph-based ConceptNet Numberbatch (CNNB). The following two paragraphs give a brief overview of these two embedding techniques.

ELMo embeddings are deep, contextualized representations of words computed using a two-layer bidirectional language model (biLM), which is pretrained on a large text corpus [28]. ELMo is robust in creating embeddings for out-of-vocabulary (OOV) words because it incorporates subword and character-level information when creating embeddings. Handling OOV words is particularly important in this work as most of the sentences often contain domain-specific keywords such as *"Glucose-6-p", "DHP", "GAP"* which can fall into the OOV category. ELMo sentence vectors are 1024-dimensional.

ConceptNet is a semantic network of knowledge about word meanings [29]. CNNB embeddings [30] are semantic word vectors created by encoding the knowledge in ConceptNet [29]. ConceptNet Numberbatch sentence embeddings are produced by taking the mean of single word

embeddings in a sentence. CNNB sentence vectors are 300-dimensional. CNNB embeddings are not as robust as ELMo embeddings when handling OOV words, yet the percentage of OOV words in the current dataset is rather small (6%). Hence we use CNNB as the second embedding technique to create sentence embeddings.

2.2.2. Implementation Details

We use Pandas DataFrames [31] to process and manipulate the sentence features. We also used other external libraries used in the extraction of sentence features as follows. To extract semantic features, the sentences are sent through a transition-based AMR parser named CAMR [32]. Textblob [33] is used to assess the subjectivity and polarity scores of the sentences. POS tag and NER-related features (in syntactic features category) are extracted using spaCy [1]. Matplotlib [2] and seaborn [3] are used for the visualizations.

ConceptNet Numberbatch embeddings are static representations for words available publicly [4]. ELMo sentence embeddings were created using the model available at Tensorflow Hub[5].

2.3. Analysis

In the following sections, we look at semantic, sentiment-based, syntactic, and statistical feature distributions for strong and weak analogies. We then compare the performances of an SVM classifier and K-means clustering.

Figure 3 illustrates the distributions of counts of concepts, relations, attributes, unique concepts, unique relations, and unique attributes of strong and weak analogies. Figure 4 presents the polarity and subjectivity distribution of strong and weak analogies. As shown in the plots, both strong and weak analogies contain sentences with neutral polarity and less subjectivity. Seventeen POS tags are present in the dataset. Distributions of the three most prevalent POS tags in strong and weak analogies are depicted in Figure 5 to utilize space effectively. Nevertheless, we used all 17 POS tags in the SVM classifier as features. Out of the fourteen named entities in the dataset (that are used in the SVM classifier), the distributions of the top three (ORG, CARDINAL, and PERSON) are plotted in Figure 6. We use the spaCy English pipeline [6] for NER tagging. Analogies written by students contain several references to biochemicals. These are misidentified as organizations (ORG) by spaCy, resulting in the ORG tag being the top named entity in the dataset. Figure 7 shows the distribution of simple and complex/ compound sentences. Weak analogies tend to have a slightly higher number of complex/ compound sentences, and strong analogies have slightly more simple sentences. Figure 8 presents the distributions of word counts, character counts, average word lengths, and unique word counts of strong and weak analogies. Modest discrepancies between distributions suggest the potential for such features to distinguish between strong and weak analogies. Therefore a feature vector combining the abovementioned features (engineered features) was then used in

[1] https://spacy.io/
[2] https://matplotlib.org/
[3] https://seaborn.pydata.org/
[4] https://github.com/commonsense/conceptnet-numberbatch
[5] https://tfhub.dev/google/elmo/3
[6] https://spacy.io/models/en#en_core_web_md

an SVM classifier to classify strong and weak analogies.
We use five variants of sentence vectors as inputs to the SVM classifier and K-means clustering. The first variant is the ELMo embeddings vector (ELMo). The second variant is the CNNB embeddings vector (CNNB), and the third variant is the engineered features vector with the vector dimension of 44. The fourth variant is a simple concatenation between ELMo embeddings and the engineered feature vector (ELMo composite). The fifth variant is a simple concatenation between CNNB embeddings and the engineered feature vector (CNNB composite).

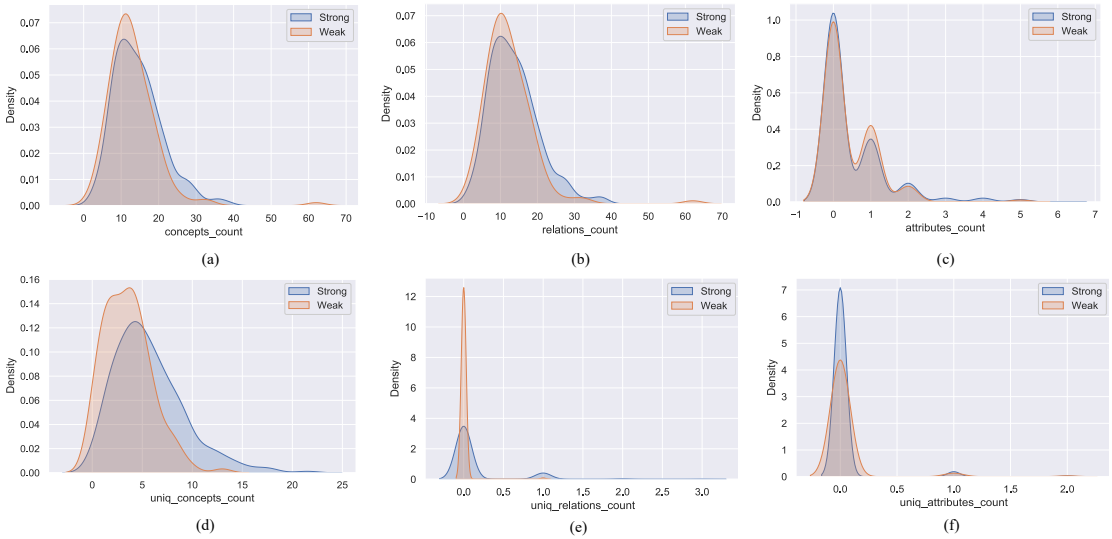

Figure 3: Plots illustrating the distributions of (a) Concepts counts, (b) Relations counts, (c) Attributes counts, (d) Unique concepts counts, (e) Unique relations counts, and (f) Unique attributes counts of sentence analogies.

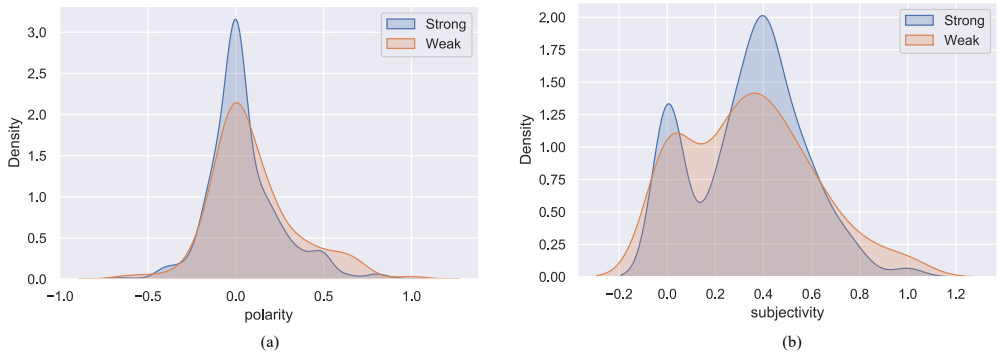

Figure 4: Plots illustrating the distributions of (a) Polarity, (b) Subjectivity across sentence analogies.

We opted to train an SVM classifier with stratified K-fold cross validation due to the limited size of our dataset (less than 1000 data points). Due to the imbalanced nature of the dataset and

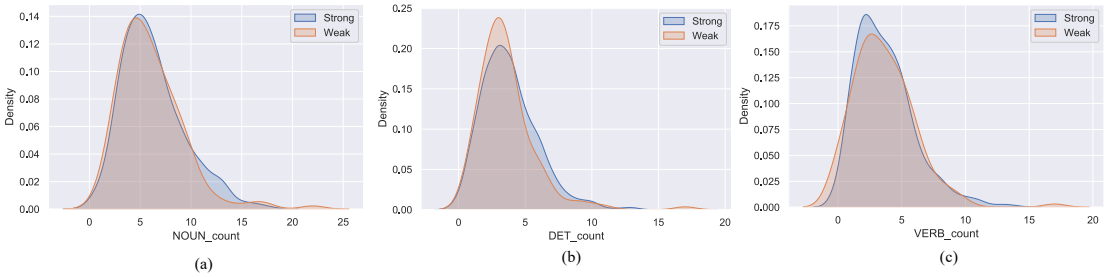

Figure 5: (a) VERB, (b) DETERMINER, (c) NOUN POS tags counts distribution across sentence analogies.

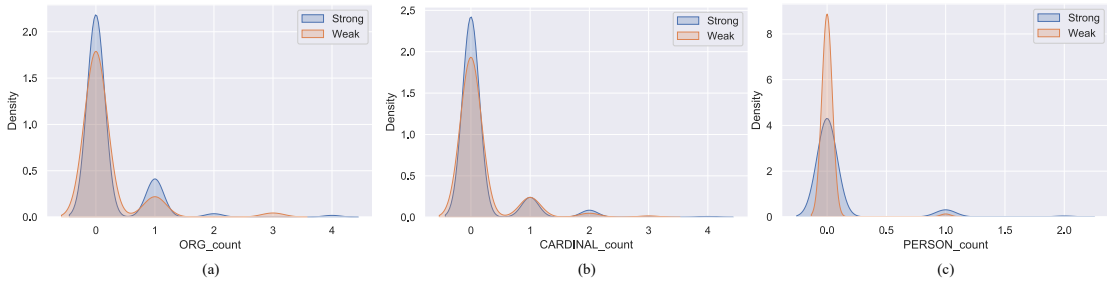

Figure 6: (a) ORG, (b) CARDINAL, and (c) PERSON NER tag counts distribution across sentence analogies.

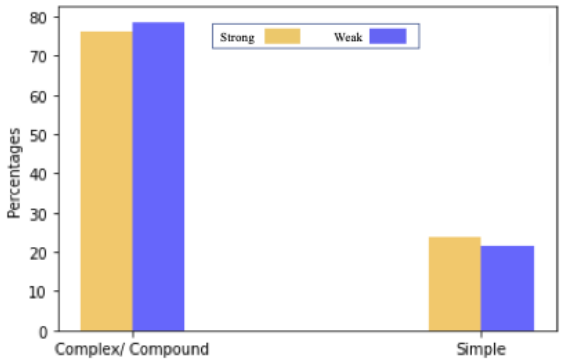

Figure 7: Sentence type statistics of strong and weak analogies.

identifying strong and weak analogies were equally important in this initial analysis, we used macro-F1 as the performance metric [34]. Performance of the SVM classifier with five variants of sentence vectors are listed in table 2.3.

We further inspect the contributions of the engineered features from the four feature categories mentioned in section 2.2 when discriminating between strong and weak analogies. We observe (see Figure 9) that most of the top 15 discriminating features belong to the syntactic

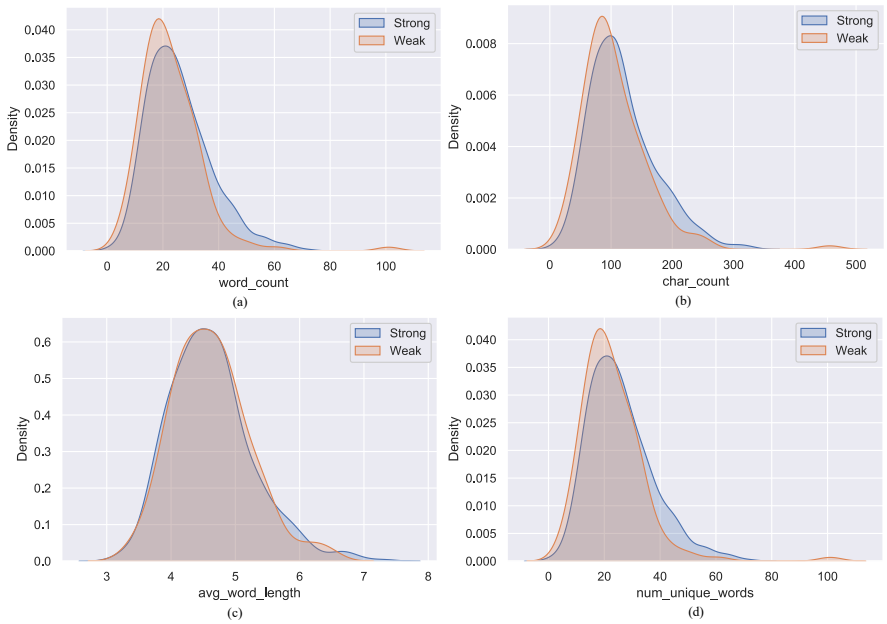

Figure 8: Plots illustrating the distributions of (a) Word count, (b) Character count, (c) Avg. word length, and (d) Number of unique words across sentence analogies.

feature category, but a semantic feature contributes the most to discriminate between strong and weak analogies.

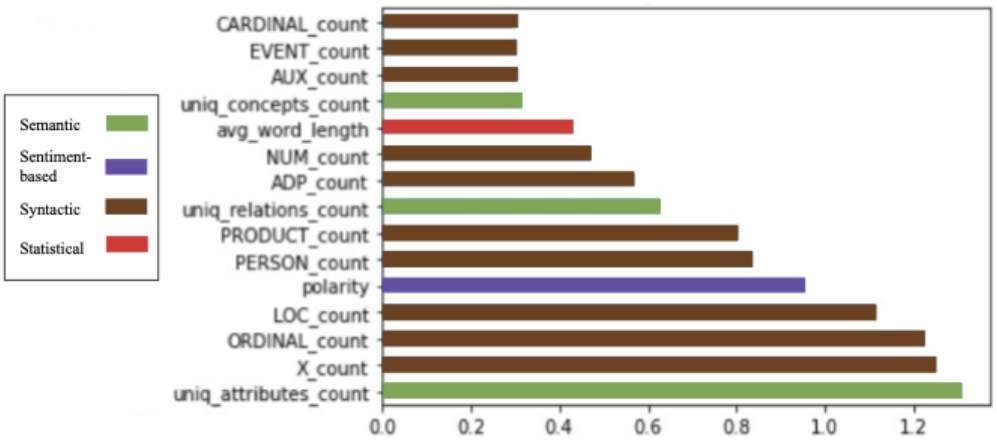

Figure 9: Top 15 discriminative features between strong and weak analogies

We use K-means to cluster the five variants of sentence vectors mentioned above with cluster centers randomly selected and K set to two. Based on the Rand index [7], the clusters are not well

[7] https://scikit-learn.org/stable/modules/clustering.html#rand-index

Table 2
Different sentence vector variants and their performance on SVM classifier measured by *macro* averaged Precision/Recall/F1-score along with their K-means cluster qualities given by Rand Index.

Vector Variant	SVM Classifier			K-Means Clustering
	Precision	Recall	F1-score	Rand Index
ELMo embeddings	0.96	0.91	0.93	0.51
CNNB embeddings	0.86	0.82	0.84	0.52
Engineered features	0.75	0.64	0.67	0.54
ELMo composite	0.96	0.96	0.96	0.50
CNNB composite	0.83	0.82	0.81	0.52

segregated in any variant, yet engineered features embeddings performed slightly better (see Table 2.3).

3. Discussion

This section presents our findings and insights, followed by the limitations and future work.

3.1. Findings and Insights

We introduce the DAF pipeline to identify discriminative features of strong and weak analogies. We show that just a few engineered features does a surprisingly good job as input to SVM. To be sure, the ELMo composite sent through the SVM classifier performs better than the rest of the sentence vector variants. Nevertheless, the ELMo composite score is slightly higher ($\sim$0.03) than the ELMo. This increase highlights that the engineered features encode some aspects of the analogies not well-captured by the ELMo embeddings. Although the SVM's performance with the engineered feature vector is 26% lower than that of the ELMo embedding, its embedding size is $\sim$23 times smaller than the ELMo. This phenomenon hints that considerable performance gains can be achieved with a much smaller number of better hand-crafted features, and most importantly, the better performance is explainable. We also note that the CNNB composite vector's performance in SVM is slightly poorer than that of the CNNB itself ($\sim$0.02). Although further exploration is required to explain this phenomenon clearly, we suspect this may be the result of feature multicollinearity specific to the manner in which CNNB creates its embeddings, combined with idiosyncracies of the subsets constructed in cross-validation.

We show that, among the features passed to the SVM classifier, the most discriminative feature for classification is a semantic feature (unique attribute count) and three out of the four semantic features (unique relations count, unique concepts count, concepts count) fall in the list of top 15 discriminative features. Also, among the top 15 discriminative features, syntactic features have the most representation. Overall, the engineered features are few in number, meaningful, and relatively cheap to calculate. Given the range of content in the data set –anything from marbles to cake–the modest success reported here is impressive. These features will contribute to our future efforts based on more computationally intensive semantic analysis. A successful unsupervised learning method would liberate classifier training from

dependence on manual annotation. Unsupervised learning results remain largely unimpressive. Nevertheless, there are some hints of promise. Engineered features improve clustering results relative to embeddings alone or embeddings and engineered features. This reinforces our claim that such features are identifying discriminators that are not captured by embeddings.

3.2. Limitations and Future Work.

The dataset used in this work is imbalanced, with more strong analogy data points than weak ones. This may cause the "uniform effect" where K-means produces clusters of the same size, even when the "true" cluster sizes of the dataset are varied [35]. To overcome such issues we plan to improve class imbalance through SMOTE [36], GANS[37], and the expansion of the manually-annotated corpus.

The natural language processing techniques employed in this work do not handle the particular nature of the dataset. For example, the spaCy model we use is trained on a generic English corpus [8]. However, we plan to use models/ techniques trained on subject-specific corpora to overcome issues like misidentifying biochemical terms as organizations in NER. Also, students use the term "like" in their target analogies to signify the similarity between their analogy and the source domain (biochemistry) concept. These are wrongly picked up by the sentiment analysis tool when evaluating polarity. Modified corpora will allow us to better manage these issues.

We classified analogy strength using individual sentences, which is both a benefit and a limitation. As a result, we identified very simple discriminators. However, some sentences in the dataset might not contribute when creating strong/ weak analogies. Constraining analysis to the sentence level requires annotation to eliminate this potential source of noise. However, the long-term goal is to evaluate analogies at the document level, for their epistemic quality. Though still vector based, our ongoing work in this area employs referent knowledge bases for both the target and variable student sources, to guide semantic interpretation.

4. Conclusion

This work introduces the DAF pipeline to identify discriminative features between strong and weak long-form analogies. We show that an SVM-based supervised-learning approach can successfully discriminate component sentences drawn from strong and weak analogies. Semantic and several syntactic features are the main contributors to discrimination, helping us to realize our goal of efficient evaluation of student generated long-form analogies.

Acknowledgments

We thank Dr. Biplav Srivastava for his valuable feedback and Dr. Nitin Jain for providing the data used in this work. We also thank the reviewers for their constructive comments.

[8] https://spacy.io/models/en#en_core_web_md

References

[1] D. Gentner, Structure-mapping: A theoretical framework for analogy, Cognitive science 7 (1983) 155–170.

[2] D. J. Chalmers, R. M. French, D. R. Hofstadter, High-level perception, representation, and analogy: A critique of artificial intelligence methodology, Journal of Experimental & Theoretical Artificial Intelligence 4 (1992) 185–211.

[3] K. J. Holyoak, P. Thagard, Mental leaps: Analogy in creative thought, MIT press, 1996.

[4] D. C. Krawczyk, R. G. Morrison, I. Viskontas, K. J. Holyoak, T. W. Chow, M. F. Mendez, B. L. Miller, B. J. Knowlton, Distraction during relational reasoning: The role of prefrontal cortex in interference control, Neuropsychologia 46 (2008) 2020–2032.

[5] R. G. Morrison, D. C. Krawczyk, K. J. Holyoak, J. E. Hummel, T. W. Chow, B. L. Miller, B. J. Knowlton, A neurocomputational model of analogical reasoning and its breakdown in frontotemporal lobar degeneration, Journal of cognitive neuroscience 16 (2004) 260–271.

[6] N. Ichien, H. Lu, K. J. Holyoak, Verbal analogy problem sets: An inventory of testing materials, Behavior research methods 52 (2020) 1803–1816.

[7] H. Prade, G. Richard, Analogical proportions: Why they are useful in ai., in: IJCAI, 2021, pp. 4568–4576.

[8] S. Lim, H. Prade, G. Richard, Classifying and completing word analogies by machine learning, International Journal of Approximate Reasoning 132 (2021) 1–25.

[9] A. Ushio, L. Espinosa-Anke, S. Schockaert, J. Camacho-Collados, BERT is to NLP what alexnet is to CV: Can pre-trained language models identify analogies?, in: ACL 2022 Workshop on Commonsense Representation and Reasoning, 2022. URL: https://openreview.net/forum?id=BdWgrMFxdW9.

[10] B. A. Spellman, K. J. Holyoak, Pragmatics in analogical mapping, Cognitive psychology 31 (1996) 307–346.

[11] B. Falkenhainer, K. D. Forbus, D. Gentner, The structure-mapping engine: Algorithm and examples, Artificial intelligence 41 (1989) 1–63.

[12] M. McLure, S. Friedman, K. Forbus, Extending analogical generalization with near-misses, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 29, 2015.

[13] R.Shrem, T. Vonderhaar, V. Shalin, N. Jain, Use of student-generated process analogies to enhance student engagement, in: (in preparation), 2022.

[14] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297.

[15] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, A. Lopez, A comprehensive survey on support vector machine classification: Applications, challenges and trends, Neurocomputing 408 (2020) 189–215.

[16] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, N. Schneider, Abstract Meaning Representation for sembanking, in: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 178–186. URL: https://aclanthology.org/W13-2322.

[17] P. Kingsbury, M. Palmer, From TreeBank to PropBank, in: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02), European Language Resources Association (ELRA), Las Palmas, Canary Islands - Spain, 2002. URL:

http://www.lrec-conf.org/proceedings/lrec2002/pdf/283.pdf.

[18] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, N. A. Smith, Toward abstractive summarization using semantic representations, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 1077–1086. URL: https://aclanthology.org/N15-1114. doi:10.3115/v1/N15-1114.

[19] M. Sachan, E. Xing, Machine comprehension using rich semantic representations, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2016, pp. 486–492.

[20] S. Rao, D. Marcu, K. Knight, H. Daumé III, Biomedical event extraction using Abstract Meaning Representation, in: BioNLP 2017, Association for Computational Linguistics, Vancouver, Canada„ 2017, pp. 126–135. URL: https://aclanthology.org/W17-2315. doi:10.18653/v1/W17-2315.

[21] L. Huang, T. Cassidy, X. Feng, H. Ji, C. Voss, J. Han, A. Sil, Liberal event extraction and event schema induction, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 258–268.

[22] C. Puschmann, A. Powell, Turning words into consumer preferences: How sentiment analysis is framed in research and the news media, Social Media+ Society 4 (2018) 2056305118797724.

[23] E. Kasmuri, H. Basiron, Subjectivity analysis in opinion mining—a systematic literature review, Int J Adv Soft Comput Appl 9 (2017) 132–159.

[24] A. Chiche, B. Yitagesu, Part of speech tagging: a systematic review of deep learning and machine learning approaches, Journal of Big Data 9 (2022) 1–25.

[25] A. Mikheev, M. Moens, C. Grover, Named entity recognition without gazetteers, in: Ninth Conference of the European Chapter of the Association for Computational Linguistics, 1999, pp. 1–8.

[26] B. Das, M. Majumder, S. Phadikar, A novel system for generating simple sentences from complex and compound sentences, International Journal of Modern Education and Computer Science 11 (2018) 57.

[27] J. MacQueen, Classification and analysis of multivariate observations, in: 5th Berkeley Symp. Math. Statist. Probability, 1967, pp. 281–297.

[28] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237. URL: https://aclanthology.org/N18-1202. doi:10.18653/v1/N18-1202.

[29] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: Thirty-first AAAI conference on artificial intelligence, 2017.

[30] R. Speer, J. Chin, An ensemble method to produce high-quality word embeddings, arXiv preprint arXiv:1604.01692 (2016).

[31] W. McKinney, et al., pandas: a foundational python library for data analysis and statistics, Python for high performance and scientific computing 14 (2011) 1–9.

[32] C. Wang, S. Pradhan, X. Pan, H. Ji, N. Xue, Camr at semeval-2016 task 8: An extended

transition-based amr parser, in: Proceedings of the 10th international workshop on semantic evaluation (semeval-2016), 2016, pp. 1173–1178.
[33] S. Loria, et al., textblob documentation, Release 0.15 2 (2018).
[34] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Information processing & management 45 (2009) 427–437.
[35] H. Xiong, J. Wu, J. Chen, K-means clustering versus validation measures: A data-distribution perspective, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39 (2009) 318–331. doi:10.1109/TSMCB.2008.2004559.
[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.
[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).