Proceedings

IJCAI-ECAI Workshop

“Interactions between Analogical Reasoning and Machine Learning”

IARML 2022

July 23, 2022

Vienna, Austria

Editors

Miguel Couceiro (University of Lorraine, CNRS, Loria)

Pierre-Alexandre Murena (Aalto University)

https://iarml2022-ijcai-ecai.loria.fr/

Preface

Analogical reasoning is a remarkable capability of human reasoning, used to solve hard reasoning tasks. It consists in transferring knowledge from a source domain to a different, but somewhat similar, target domain by relying simultaneously on similarities and dissimilarities. In particular, analogical proportions, i.e., statements of the form "A is to B as C is to D", are the basis of analogical inference. Analogical reasoning is pertaining to case-based reasoning and it has contributed to multiple machine learning tasks such as classification, decision making, and automatic translation with competitive results. Moreover, analogical extrapolation can support dataset augmentation (analogical extension) for model learning, especially in environments with few labeled examples. Conversely, advanced neural techniques, such as representation learning, enabled efficient approaches to detecting and solving analogies in domains where symbolic approaches had shown their limits. However, recent approaches using deep learning architectures remain task and domain specific, and strongly rely on ad-hoc representations of objects, i.e., tailor made embeddings.

The first workshop *Interactions between Analogical Reasoning and Machine Learning* (IARML) is being hosted by the 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI-ECAI 2022). It brings together AI researchers at the cross roads of machine learning, cognitive sciences and knowledge representation and reasoning, who are interested by the various applications of analogical reasoning in machine learning or, conversely, of machine learning techniques to improve analogical reasoning. The IARML workshop aims to bridge gaps between different AI communities, including case-based reasoning, deep learning and neuro-symbolic machine learning. The workshop welcomed submissions of research papers on all topics at the intersection of analogical reasoning and machine learning. The submissions were subjected to a strict double-blind reviewing process that resulted in the selection of six original contributions and two invited talks, in addition to the two plenary keynote talks.

Invited talks:

Towards a Model of Visual Reasoning (Ekaterina Y. Shurkova*, Leonidas Doumas)

Analogical Proportions (Christian Antic)

Plenary talks:

Analogy as a Technology for Machine Learning (Kenneth Forbus)

Analogy on Text Data (Yves Lepage)

IARML@IJCAI-ECAI'22 took place on July 23, 2022 in Vienna (Austria), and we are truly thankful to the IJCAI-ECAI workshop chairs for their help in the organization of this event. We are greatly indebt to the scientific committee for their reviews and suggestions for improving the accepted contributions.

Miguel Couceiro
Pierre-Alexandre Murena

Organising Committee

Miguel Couceiro (University of Lorraine, CNRS, Loria, FR)

Pierre-Alexandre Murena (Aalto University, FI)

Scientific Committee

Stergos Afantenos (CNRS, Université Paul Sabatier, IRIT, FR)

Fadi Badra (Université Sorbonne Paris Nord, LIMICS, FR)

Nelly Barbot (Université de Rennes 1, IRISA, FR)

Tarek R. Besold (DEKRA DIGITAL, Eindhoven University of Technology, NL)

Myriam Bounhas (LARODEC-ISGT, TU, UAE)

Adrien Coulet (Inria Paris, FR)

Sebastien Destercke (CNRS, Université de Technologie de Compiegne, Heudiasyc, FR)

Claire Gardent (University of Lorraine, CNRS, LORIA, FR)

Eyke Hullermeier (University of Munich, DE)

Mehdi Kaytoue (Infologic, FR)

Yves Lepage (Waseda University, JA)

Jean Lieber (University of Lorraine, CNRS, LORIA, FR)

Esteban Marquer (University of Lorraine, CNRS, LORIA, FR)

Laurent Miclet (Université de Rennes, FR)

Pierre Monnin (Orange, FR)

Amedeo Napoli (University of Lorraine, CNRS, LORIA, FR)

Henri Prade (CNRS, Université Paul Sabatier, IRIT, FR)

Irina Rabkina (OXY Occidental College, USA)

Steven Schockaert (Cardiff University, IR)

Table of Contents

Accepted papers

Masked prompt learning for formal analogies beyond words

Liyan Wang[1,*], Yves Lepage[1]

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

Abstract

Prompt learning, a recent thread in few-shot learning for pre-trained language models (PLMs), has been explored for completing word analogies in the extractive way. In this paper, we reformulate the analogy task as masked analogy completion task with the use of prompting to derive a generative model for analogies beyond words. We introduce a simple prompt-based fine-tuning paradigm for language modeling on answered prompts of analogies in the sequence-to-sequence framework. To convert discrete terms of analogies into linear sequences, we present a symbolic prompt template. The sequence-to-sequence model is fine-tuned to fill in the missing span of masked prompts deduced from different masking schemes on phrase analogies extracted from a small corpus. We analyze the out-of-distribution performance on sentence analogies which are unseen cases. Our experiments demonstrate that prompt-based fine-tuning with the objective of language modeling enables models to achieve significantly better performance on in-distribution cases than PLMs. Masked prompt learning with one-term masking exhibits the best out-of-distribution generalization on sentence analogies, with a difference of only 3 characters from references.

Keywords

Prompt learning, masked analogy completion, analogies beyond words, fine-tuning

1. Introduction

Analogy, a cognition mechanism that relies on relational similarity, is growing in prominence in the field of artificial intelligence [1]. In general, it encapsulates a quadruplet relationship between terms of the same type. For example, the famous analogy between words *king* : *queen* :: *man* : *woman*, implies that *king* is to *queen* as *man* is to *woman* in terms of gender transition. Strikingly, analogical quadruples form geometric parallelograms in pre-trained embedding spaces learnt by the skip-gram model with negative sampling [2]. The parallelogram generic has drawn attention in research about the linear algebraic properties of vector analogies [3]. The simple arithmetic approach, however, was questioned as being applicable only for completing analogies with respect to certain clearly defined relations [4]. Recent efforts [5, 6] have examined the potential of machine learning techniques to learn analogies in word embedding spaces.

By formulating tasks in the manner of analogical reasoning, sentence analogy has demonstrated its versatility in various tasks in the area of natural language processing (NLP), such

IARML@IJCAI-ECAI'2022: Workshop on the Interactions between Analogical Reasoning and Machine Learning, at IJCAI-ECAI'2022, July, 2022, Vienna, Austria

[*] Corresponding author.

✉ wangliyan0905@toki.waseda.jp (L. Wang); yves.lepage@waseda.jp (Y. Lepage)

CEUR Workshop Proceedings (CEUR-WS.org)

as machine translation [7], text summarization [8], and question answering [9]. In contrast to word-level analogies, analogies that go beyond words may encompass manifold challenges that stem from the inherent complexity of language. A recent work [10] has demonstrated that sentence embedding models struggle to capture analogical regularities in terms of geometric parallelism. The vector offsets may not be kept within sentence embeddings. In [11], the postulate of exchange of the means has been debated for classifying positive and negative analogies between sentences.

Some work on completing sentence analogies has focused on extractive approaches, i.e., identifying the optimum solution from candidates (i.e., from a finite answer pool) [9, 8]. However, the extractive mechanism is not geared to text generation. It is relatively expensive to define candidate sets for analogies that capture complex relations between long sequences. In addition, hand-crafted candidates may leave weaknesses in linguistic creativity. Therefore, it highlights the necessity for a generative model that can automatically produce the missing term in analogy questions, which will contribute to analogy completion in NLP scenarios.

Prompt learning [12] is a fruitful learning paradigm in recent work on the adaptive performance of PLMs in the few-shot setting. The downstream tasks are reformulated as Cloze-style problems by converting inputs into natural language prompts with task-specific descriptions, which allows PLMs to predict target outputs conditioned on prompts [13]. Following [13], a filled prompt refers to a prompt where the mask slot is filled with any answer. An answered prompt refers to a prompt filled with the correct answer. Based on prompt learning, recent works have investigated the few-shot [12] and zero-shot [14] performance of PLMs in identifying word analogies from candidates by language model (LM) scoring on filled prompts.

In this paper, we present a preliminary study of generative completion of analogies beyond words by using LMs in conjunction with prompting. To learn analogical regularities, we introduce a novel prompt-based fine-tuning method to extract sequential features of answered prompts with the symbolic template by masked sequence-to-sequence learning. We conduct lightweight fine-tuning on phrase analogies with different masking schemes. By language modeling on answered prompts, fine-tuned models achieve over 97% accuracy on solving phrase analogies. The fine-tuned sequence-to-sequence LMs have more promising out-of-distribution generalization on sentence analogies than autoregressive LMs. In particular, one-term masking is more robust at extracting analogical regularities, which contributes to adapt effectively to unseen analogies.

2. Related Work

Based on prompting, recent works have formulated the quadruplet problem as language modeling. The GPT-3 work [12] explored the adaptation of the explosive-size LM (with 175B parameters, which is over 100 times larger than GPT-2) on Stochastic Aptitude Tests (SAT) analogy task in the few shot setting with no gradient updates. The discrete texts in analogies are mapped into natural sentences with a textual prompt template. The pre-trained GPT-3 model learns within the context consisting of few answered prompts and the query, to infer the answer with the maximum LM likelihood among the given candidates. This exhibits the potential of GPT-3 for extractive analogy completion at the word level. It gave rise to the exploration of analogical capabilities of LMs with the help of prompts. A recent work [14] examined the adaptability of PLMs to recognize word analogies with different levels of complexity in the zero-shot setting. They conduct ablation experiments in several aspects including prompt engineering, architecture engineering and scoring engineering. With the appropriate choices, PLMs can achieve meaningful zero-shot performance on analogy identification.

Recent efforts on prompt learning found that prompt-based fine-tuning of PLMs can improve effective learning on downstream tasks. These works have the advantage of being specific to the task by tuning the LM parameters entirely [15] or partially [16] on a small number of examples. Typically, prompt-based fine-tuning is adopted on masked language models (MLMs) to predict the mask token fixedly pointing to the target label in prompts. Here, we graft prompt-based fine-tuning onto a sequence-to-sequence LM with different masking schemes and execute masked sequence-to-sequence learning on a large number of examples to extensively model prompt sequences. The analogy task is reformulated as masked analogy completion, where sequence answers are generated by a fine-tuned LM to fill in the mask token in prompts for analogy questions.

3. Method

In this section, we introduce a generative method for completing analogies beyond words by using prompts with a symbolic template (Section 3.1). To adapt to the analogy task, we propose a novel prompt-based fine-tuning paradigm (Section 3.3) for PLMs to reconstruct the missing span in prompts processed by masking schemes (Section 3.2).

3.1. Symbolic Prompt Template

Prompt design is crucial in prompt learning. For analogies beyond words, it is easy to get the tokens of analogical words mixed up with the context tokens of textual templates, as used in [12, 14]. We introduce a clear prompt in which the contents of the four terms are easily parsed out from sequences.

Symbolic prompts for analogy are formally identical to analogical equations $A : B :: C : D$, where the ratio (:) and proportion (::) characters are two symbolic tokens in the prompt template. We employ the Unicode characters `U+2236` and `U+2237` for ratio and proportion in sequences. Let X ($X \in \{A, B, C, D\}$) denote one of the four terms in an analogy. The length of an answered prompt can be calculated as $|A| + 1 + |B| + 1 + |C| + 1 + |D| = \sum_{X \in \{A,B,C,D\}} |X| + 3$.

Compared to textual tokens, symbolic tokens that facilitate the distinction between the four terms are straightforward. Based on ordering characteristics, they directly delimit the four terms by detecting the order of symbolic tokens in sequences. For example, : and :: delimit the term B, whereas :: and :, in that order, delimit the term C.

3.2. Masking Schemes

In light of the usual notation for analogies $A : B :: C : x,$ it is natural to consider the expected solution x as the missing text that can be predicted using the left context. To learn sequential information, we explore three patterns of masking for answered prompts. We present some examples of masked sequences that result from masking the following sentence analogy to exemplify masking schemes.

he will come tomorrow. : *he will come.* :: *i have no time tomorrow.* : *i have no time.*

Arbitrary Masking (Mask = any span) Like the document corruption method introduced in [17], we randomly mask consecutive tokens whose lengths follow a sampling distribution $\lambda = \mathit{Poisson}(3)$. This masking strategy does not take into account the structure of analogies. The starting position of each masking span is selected at random. A masking span can consist of a symbolic token and parts of adjacent terms. Like the following resulted sequence, part of the first two terms are masked off along with the left ratio token.

he will [mask] *will come.* :: *i have no time tomorrow.* : *i have no time.*

One-term Masking (Mask = any term) A masking span is a whole term in analogies, selected from the quadruplet (A, B, C, D). Due to the binding meaning between the four terms in an analogy, each term can be derived from the other three. This scheme randomly masks one of the terms. It allows models to capture more comprehensively analogical regularities. The term C is masked in the following sequence.

he will come tomorrow. : *he will come.* :: [mask] : *i have no time.*

Target-oriented Masking (Mask = term *D*) In this setting, we regard the target prediction as the masking span in analogy prompts. To follow standard notations in analogy, i.e., the format *A* : *B* :: *C* : *x*, we specifically mask the fourth term in answered prompts as shown below.

he will come tomorrow. : *he will come.* :: *i have no time tomorrow.* : [mask]

3.3. Masked Prompt Learning

In general, the paradigm of prompt-based fine-tuning reformulates downstream tasks as masked language modeling on prompts, with the goal of optimizing the prediction of the mask token specified as target outputs [16, 15]. For text generation, we introduce a novel prompt-based fine-tuning paradigm to extract sequential information of answered prompts by masked sequence-to-sequence learning like [18]. On this basis, the analogy task is formulated as masked analogy completion, which aims to generate unknown terms through a sequence-to-sequence model trained for reconstructing the masked span in prompts.

Given a pair of sequences (X, Y) , where X is the sequence of a masked prompt (including a single mask token) obtained by applying a masking scheme to the answered prompt, Y is the target sequence of the masking span. As in regular sequence-to-sequence learning, we tune the model parameters $\Theta = (\theta_{enc}, \theta_{dec})$ to estimate the conditional probabilities of target tokens given masked prompts. The masked prompt is encoded bidirectionally. Each token in the target sequence is predicted by the autoregressive decoder by maximizing the conditional probability given the input sequence x and its preceding sequence $Y_{<t}$:

$$P\left(Y\left|X\right.\right)=\prod_{t=1}^{|Y|}P\left(Y_t\left|Y_{<t},X;\Theta\right.\right) \tag{1}$$

The loss function for reconstructing a masked prompt is computed as the negative log-likelihood (Equation 2). Our training objective is to minimize the loss function, which is tantamount to maximizing conditional probabilities.

$$L_{\mathrm{m}}\left(X,Y\right)=-\log P\left(Y\left|X\right.\right) \tag{2}$$

The mechanism of masked prompt learning with target-oriented masking scheme (term *D*), resembles few-shot prompt-based fine-tuning for MLMs, which focuses on learning relations between prompts and target outputs of downstream tasks. We will compare the fine-tuning paradigms under different masking schemes in Section 5.3.

4. Formal Analogy between Phrases

Note that finding sentence analogies from a corpus will be difficult unless the corpus is dense. It is easier to find more analogies between small chunks from a corpus than entire sentences. Sentence constituents (i.e., phrases) are structural chunks that play grammatical roles in sentences. We focus on finding formal analogies between phrases, which can indirectly reflect the linguistic regularities contained in the given corpus.

We build analogies from the English part of a parallel corpus[1] released for the news translation task at the Workshop on Machine Translation (WMT20). The data we used is made up of 3,003 sentences with an average length of 25 words. In order to detect phrases, we use Berkeley Neural Parser[2][19] to parse sentences into constituency trees. In each sentence, word sequences between 2 and 6 in length are collected into a phrase pool from which analogies are identified. After traversing all sentences, we obtain 25,310 phrases.

Starting from the set of phrases, we apply some functions from the Nlg tool[3] introduced in [20] to extract analogies. In this work, we explore analogical relations at the formal level. Each phrase is represented as a bag-of-word vector using *Lines2Vectors*. The dimension of vectors is the number of word types in the phrase set. We then run *Vectors2Clusters* to find analogical clusters pertaining to the differences between phrase vectors. Each cluster is made up of pairs of phrases (i.e. ratios) that have the same syntactic transformations. Thus, any two ratios in a cluster makes a syntactic analogy. Note that cluster sizes may vary greatly depending on frequencies of phrase structures contained in the corpus. To alleviate the imbalance of possible analogies, we set the maximum cluster size to 10, where the minimum size defaults to 2.

In our settings, over 1.5 million analogical clusters are extracted, where each cluster contains two phrase ratios in average. By combining every two ratios in a cluster, we are able to enumerate

[1] It can be downloaded from http://www.statmt.org/wmt20/translation-task.html

[2] https://github.com/nikitakit/self-attentive-parser

[3] http://lepage-lab.ips.waseda.ac.jp/en/projects/kakenhi-15k00317/ → Tools - Nlg Module

1,524,293 analogies that capture formal similarities between four distinct phrases. The analogy data includes 17,480 types of phrases with an average length of 3 words (20 characters). It can be roughly estimated that the average length of analogy prompts is $3 \times 4 + 3$ in words (resp. $20 \times 4 + 3$ in characters). For example, a collected phrase analogy *to say* : *want to say* :: *to go out* : *want to go out* indicates a verb phrase attachment with the modifier of the verb *want*, which consists of 15 words including three symbolic tokens in the prompt template.

5. Experiments

5.1. Datasets

We fine-tune LMs on prompts of phrase analogies. To avoid that phrase ratios in the test data appear also in the training set, we split the cluster data before making analogies. We take 1,000 clusters for building the analogies for testing and another 1,000 for validation, while the remaining ones are for training. For each cluster set, we assemble every two ratios in the same cluster into an analogy. The training, validation and test sets consist of approximately 1.5 million, 1,000 and 1,000 phrase analogies respectively.

In addition to the phrase analogy test set, we also explore the performance of Transformer models on solving sentence analogies, which are unseen analogies and have different distributions than the training data. We sample sentence analogies from the English part of bilingual analogies[4] used in an EBMT by analogy setting [21]. The set of analogies embraces formal-level analogous relationships between sentences from the Tatoeba corpus, where the length of sentences varies from 2 to 10.

Even if we swap the ordering of four phrases, each analogy only appears once in all datasets, with no duplicates. The length statistics of analogies are shown in Table 1, including lengths of terms and answered prompts. Analogies in the test sets are processed as prefix prompts, in which the mask token is the last token in sequences. Each model is tested by infilling the unknown term in masked prompts with the default format $A : B :: C :$ [mask].

5.2. Training Details

In terms of the sequence-to-sequence Transformer architecture, we experiment with a pre-trained BART [17]. For computational efficiency, we use the base-size model consisting of a 6-layer bidirectional encoder and a 6-layer autoregressive decoder. The pre-training paradigm of BART is to perform denoising learning on corrupted text, to reconstruct the entire completed sequences. To remove duplicates that to reconstruct the known tokens of masked sequences, we fine-tune BART through masked prompt learning, which reconstructs only the sequences of masking fragments.

We made some modifications to the BART fine-tuning procedure provided by the *Transformers* library [22]. To save computational memory, we conducted partial freezing on the pre-trained BART model and fine-tune only four of the twelve layers, precisely the bottom two layers of the encoder and the top two layers of the decoder. We analyze the discrepancies on fine-tuning

[4] The bilingual set of sentence analogies is the 3rd resource of experimental results at http://lepage-lab.ips.waseda.ac.jp/en/projects/kakenhi-kiban-c-18k11447/.

Table 1

Statistics of span lengths in prompting sequences including four terms and answered prompts in two analogy data: phrase analogies (PA) and sentence analogies (SA). We split over 1.5 million phrases analogies into three sets for training (1.5 million), validation (1,000) and test (1,000). In addition, we also test models on 1,000 sentence analogies which are out-of-distribution cases.

(a) Phrase analogies (PA)

Span	Length in words	 in chars
A	2.77±0.00	17.07±0.02
B	3.09±0.00	18.82±0.02
C	3.71±0.00	20.03±0.02
D	4.04±0.00	21.78±0.02
Prompt	16.61±0.01	80.69±0.05

(b) Sentence analogies (SA)

Span	Length in words	 in chars
A	5.19±0.11	19.07±0.48
B	4.82±0.09	17.48±0.38
C	5.59±0.11	20.76±0.50
D	5.22±0.11	19.17±0.47
Prompt	23.81±0.32	79.47±1.42

strategies in Section 5.5. In order to alleviate overfitting, we stop the training if there is no improvement on the metric of the validation set after two consecutive epochs. We then save the model with the best performance among the checkpoints.

5.3. Main Results

As for the autoregressive baseline, we explore the performance of the distilled GPT-2 model consisting of 6 layers. The entire pre-trained GPT-2 is fine-tuned on answered prompts of phrase analogies for generating the last term given preceding context. To compare different methods, we employ the accuracy metric to measure the percentage of generations that exactly match the references. In addition, we also compute the Levenshtein distance (including spaces) to measure differences at the character level. Table 3 shows the performance of different fine-tuned models on completing phrase analogies and sentence analogies.

Model Architecture Prompting together with fine-tuning, benefits PLMs to accomplish effective analogy completion of in-distribution cases (phrase analogies). As far as the model architecture is concerned, the autoregressive LM excels in inferring answers to phrase analogy questions where the last term (D) is missing. By learning on answered prompts of phrase analogies in a feed-forward fashion, GPT-2 achieves the best accuracy 99.7% on phrase analogies. However, the sequence-to-sequence model fine-tuned for infilling the term D performs noticeably worse, only reaching 50.9% in accuracy. Except for the setting of target-oriented masking (term D), BART fine-tuned with masked prompt learning have competitive performance with GPT-2.

Masking Scheme The masking deployment of prompts has a large impact on capturing analogical regularities in masked prompt learning for the sequence-to-sequence model. The target-oriented masking scheme (term D), like the mechanism of few-shot prompt-based fine-tuning in MLMs, performs the worst in phrase analogies. We posit that it makes BART overly

Table 3
Comparison between the autoregressive baseline and masked prompt learning with different masking schemes over two analogy test sets.

Data	Model	Masking scheme	Acc (%)	Levenshtein distance in chars
PA	GPT-2	-	**99.7**	**0.01±0.01**
	BART	Any span	97.5	0.05±0.03
		Any term	97.0	0.05±0.03
		Term *D*	50.9	0.58±0.07
SA	GPT-2	-	4.2	12.92±0.83
	BART	Any span	11.4	4.28±0.38
		Any term	**44.4**	**2.85±0.29**
		Term *D*	12.0	3.17±0.30

imitate the features of the fourth terms of analogies rather than comprehend analogical relationships. The arbitrary masking (any span) and the one-term masking (any term) are optimal for modeling the sequential information of prompts for phrase analogies, which enables the model to perform well in generating the last phrase in analogies, with only a 0.05 character difference.

Out-of-distribution Generalization As shown in the results of SA in Table 3, the autoregressive LM exhibits poor out-of-distribution generalization capability, although it achieves excellent performance on phrase analogies. GPT-2 can only correctly answer 4.2% unseen sentence analogies where sequences are longer than the training data. This suggests that the autoregressive LM may overfit to the narrow distribution of phrase analogies, which leads to a failure in solving analogies between longer sequences. In comparison, the sequence-to-sequence models exhibit better out-of-distribution generalization. Particularly, BART with similar fine-tuning procedure, predicting the term D conditional on previous tokens, is approximately three times superior to GPT-2.

It is noticeable that masked prompt learning on prompts with one-term masking (any term) has the best generalization on sentence analogies. It is 4 times more accurate than any span masking with competitive performance on phrase analogies. It enables BART to generate sequences that very closely match the reference sentences, differing by only 3 characters on average, about 3/20 of the reference length. Table 5 presents some example errors. The fine-tuned BART model profiting from the bidirectional learning on previous and future tokens, can accomplish effective adaptation to unseen sentence analogies, with the achievement of an order of magnitude greater accuracy than GPT-2.

5.4. Fine-grained Exploration of Analogical Ability

To further explore analogical capabilities of LMs fine-tuned by masked prompt learning, we conduct a fine-grained probing on analogical questions with different formats. For each test analogy, we replace one of the four terms with the mask token to enumerate four analogy questions with different masking spans. We use an individual accuracy Acc_X (where $X \in \{A, B, C, D\}$) to indicate the performance in solving analogies in a specific format. For example,

Table 4
Average accuracy results of fine-tuned BART models with different masking schemes in solving analogy questions in different formats. We report both individual accuracies Acc$_X$ (where $X \in \{A, B, C, D\}$) and universal accuracy Acc$_{\text{all}}$ introduced in Subsection 5.4 .

Data	Masking Scheme	Acc (%)				
		A	B	C	D	all
PA	Any span	**99.9**	**99.9**	**99.7**	**97.5**	**97.3**
	Any term	99.8	99.7	**99.7**	97.0	96.5
	Term *D*	0.0	0.0	0.0	50.9	0.0
SA	Any span	10.8	10.6	15.7	11.4	0.6
	Any term	**24.9**	**39.3**	**44.0**	**44.4**	**12.9**
	Term *D*	1.2	0.0	0.0	12.0	0.0

Acc$_A$ is the accuracy of answering masked prompts with the missing term A ($[\text{mask}] : B :: C : D$). Apart from individual accuracies, we also compute the universal score Acc$_{\text{all}}$ as Equation 3 to compare the performance for correctly answering all of the four masked prompts for each analogy.

$$\text{Acc}_{\text{all}} = \begin{cases} 1, & \text{if Acc}_X = 1 \text{ for all } X = A, B, C, D. \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Table 4 shows the accuracy results of fine-tuned BART models on phrase analogies and sentence analogies. For the results of phrase analogies, the masking of any span has the best overall accuracy, with 97.3% of analogies being answered correctly on each term. Among the mask settings that take analogical terms into account, the masking scheme of any term achieves competitive performance in terms of superior accuracy on each individual question, while term-specific masking enables the model to answer only half of the questions where term D is masked. For performance on sentence analogies, fine-tuning for any term masking outperforms substantially other strategies in terms of both individual accuracies and universal accuracy. Concretely, the fine-tuned model performs relatively well on completing sentence analogies where one of the last triplets is missing. It is not a surprise that BART with the target-oriented masking (term *D*) fails to predict terms other than the term *D*. The reason is that the mask token does not appear in any position during the fine-tuning procedure except for the last term.

5.5. Ablation Studies

The masking scheme of any term is more adapted to solving analogies in masked prompt learning. In this subsection , we use one-term masking (any term) and ablate fine-tuning strategies of tuned layers and prompt templates.

Fine-tuned Layers As shown in Table 6, we compare various fine-tuning strategies with the off-the-shelf baseline. In the zero-shot setting, the pre-trained model is not able to generate a reliable answer for analogies beyond words. Fine-tuning BART makes a significant impact on understanding the analogical relationships between sequences. We can see that updating the

Table 5
Examples of analogy results generated by the BART fine-tuned with one-term masking. The underlined text is the difference between the output and the reference.

Input	*he is getting better.* : *tom is getting better.* :: *he doesn 't like to lose.* : [mask]
Output	*tom doesn 't like to lose*
Reference	*tom doesn 't like to lose.*
Input	*i know your name.* : *i believe you.* :: *i don 't know your name.* : [mask]
Output	*don 't believe you.*
Reference	*i don 't believe you.*
Input	*what is going on here?* : *he is intelligent.* :: *what's going on here?* : [mask]
Output	*he is intelligent.*
Reference	*he's intelligent.*
Input	*how did you do this?* : *what did you say?* :: *how do you do this?* : [mask]
Output	*what did you say?*
Reference	*what do you say?*
Input	*he will come tomorrow.* : *he will come.* :: *i have no time tomorrow.* : [mask]
Output	*he will have no time tomorrow.*
Reference	*i have no time.*

entire model helps model phrase analogies, with a significant gain of 94.2 points in predicting the last phrase in analogy questions.

However, it is imprudent to update the entire model. The full-scale fine-tuning makes the model specialized in the training distribution, achieving only 11.7% accuracy in completing sentence analogies. Freezing the entire encoder or decoder degrades the performance of solving phrase analogies, while it increases the performance by at least 12 points over the unfrozen BART for generalizing out-of-distribution analogies. In particular, only tuning the decoder and freeze the encoder is useful to learn phrase distributions, whereas fine-tuning the encoder and freeze the decoder performs relatively better for capturing analogical regularities.

By contrast, lightweight joint fine-tuning on both encoder and decoder performs well on two analogy test sets. Fine-tuning the bottleneck layers (the top two layers of encoder and the bottom two layers of decoder), which closely updates the encoder-decoder attention, achieves accuracy scores of 95.8% and 42.0% on phrase and sentence analogies respectively. Our strategy of fine-tuning only the bottom two layers of the encoder and the top two layers of the decoder, exhibits the best performance, yielding slight gains of about 2 points over fine-tuning the bottleneck layers of BART.

Prompt Templates In Table 7, we list results of GPT-2 and BART learned on prompts with different manually written templates including symbolic prompt, textual prompt and null prompt.[5] As findings in [16], different manually-written templates lead to similar in-distribution accuracy. However, prompt templates behave differently on out-of-distribution cases. We can

[5] We also perform experiments on pre-trained GPT-2 and BART. Regardless of the template, almost none of the analogies can be answered correctly by PLMs in the zero-shot setting.

Table 6
Impact of fine-tuning strategies of fine-tuned layers in the pre-trained sequence-to-sequence model.

Fine-tuned layer		Acc (%)	
Encoder	Decoder	PA	SA
None	None	0.0	0.0
All	All	94.2	11.7
None	All	88.6	23.7
All	None	70.5	28.7
Bottom two	Top two	**97.0**	**44.4**
Top two	Bottom two	95.8	42.0

Table 7
Comparison between different prompt templates: symbolic prompt ($A : B :: C : D$), textual prompt (*A is to B as C is to D*), and null prompt (*A B C D*).

Template	Model	Acc (%)	
		PA	SA
$A : B :: C : D$	GPT-2	**99.7**	4.2
	BART	97.0	**44.4**
A is to B as C is to D	GPT-2	**99.9**	3.6
	BART	99.6	**37.9**
A B C D	GPT-2	**99.9**	0.0
	BART	99.6	0.0

observe that models fine-tuned on simple concatenation of four analogical phrases (null prompt) are not able to answer sentence analogies, although they achieve 99.9% accuracy on phrase analogies. The accuracy of non-null prompts (textual and symbolic templates) is increased by at least 37.9% on sentence analogies.

Despite a subtle drop (1.4 points) in the phrase analogy test, our prompt contributes to better adaptation on unseen analogies than the textual prompt used in [14], attaining an improvement of 4.5 points for BART (0.6 point for GPT-2). This shows the necessity of clear prompt semantics, which allow models to better learn analogical relations encapsulated in prompts.

Regardless of the template, GPT-2 models struggle in completing sentence analogies, although they excel in completing phrase analogies. In contrast, the BART model is over 10 times more accurate than GPT-2 in sentence analogies with the help of non-null prompts.

6. Conclusion

Our work demonstrated the potential of LMs to complete analogies beyond words. We introduced a simple but effective prompt-based fine-tuning paradigm for solving analogies beyond words by masked sequence-to-sequence learning on answered prompts with different masking schemes. To extract useful information about analogical regularities, we proposed three patterns of masking on answered prompts. We found that fine-tuning with the objective of language modeling on answered prompts, is effective for adaptation of generative analogy completion on phrase analogies, except the sequence-to-sequence framework with target-oriented masking which leads to overfit to narrow features in the training data. Compared to the autoregressive framework, masked prompt learning is beneficial for out-of-distribution generalization over sentence analogies. Lightweight fine-tuning in masked prompt learning with one-term masking has the best potential for learning robust analogical capabilities. In the future, we intend to refine the fine-tuning paradigm to enhance out-of-distribution performance in the few-shot scenario. We hope to apply to other languages and build a multilingual generator for analogies beyond words.

Acknowledgments

The work is supported by China Scholarship Council (CSC) under the CSC Grant No. 202008050136.

References

[1] H. Prade, G. Richard, Analogical proportions: Why they are useful in ai, in: Z.-H. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, International Joint Conferences on Artificial Intelligence Organization, 2021, pp. 4568–4576. doi:10.24963/ijcai.2021/621, survey Track.

[2] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Curran Associates Inc., Red Hook, NY, USA, 2013, p. 3111–3119.

[3] O. Levy, Y. Goldberg, Linguistic regularities in sparse and explicit word representations, in: Proceedings of the Eighteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, Ann Arbor, Michigan, 2014, pp. 171–180. doi:10.3115/v1/W14-1618.

[4] Z. Bouraoui, S. Jameel, S. Schockaert, Relation induction in word embeddings revisited, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 1627–1637. URL: https://aclanthology.org/C18-1138.

[5] S. Lim, H. Prade, G. Richard, Classifying and completing word analogies by machine learning, International Journal of Approximate Reasoning 132 (2021) 1–25. doi:https://doi.org/10.1016/j.ijar.2021.02.002.

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

[7] Y. Lepage, E. Denoual, Purest ever example-based machine translation: Detailed presentation and assessment, Machine Translation 19 (2005) 251–282. doi:10.1007/s10590-006-9010-x.

[8] B. Elayeb, A. Chouigui, M. Bounhas, O. B. Khiroun, Automatic Arabic text summarization using analogical proportions, Cognitive Computation 12 (2020) 1043–1069. doi:10.1007/s12559-020-09748-y.

[9] A. Diallo, M. Zopf, J. Fürnkranz, Learning analogy-preserving sentence embeddings for answer selection, in: Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 910–919. doi:10.18653/v1/K19-1085.

[10] X. Zhu, G. de Melo, Sentence analogies: Linguistic regularities in sentence embeddings, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 3389–3400. doi:10.18653/v1/2020.coling-main.300.

[11] S. 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, Springer International Publishing, Cham, 2021, pp. 3–18.

[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[13] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, arXiv preprint arXiv:2107.13586 (2021).

[14] 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: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 3609–3624. doi:10.18653/v1/2021.acl-long.280.

[15] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 3816–3830. doi:10.18653/v1/2021.acl-long.295.

[16] R. Logan IV, I. Balazevic, E. Wallace, F. Petroni, S. Singh, S. Riedel, Cutting down on prompts and parameters: Simple few-shot learning with language models, in: Findings of the Association for Computational Linguistics: ACL 2022, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2824–2835. doi:10.18653/v1/2022.findings-acl.222.

[17] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of

the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. doi:10.18653/v1/2020.acl-main.703.

[18] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, MASS: Masked sequence to sequence pre-training for language generation, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 5926–5936. URL: https://proceedings.mlr.press/v97/song19d.html.

[19] N. Kitaev, D. Klein, Constituency parsing with a self-attentive encoder, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 2676–2686. doi:10.18653/v1/P18-1249.

[20] R. Fam, Y. Lepage, Tools for the production of analogical grids and a resource of n-gram analogical grids in 11 languages, in: LREC 2018, Miyazaki, Japan, 2018. URL: https://www.aclweb.org/anthology/L18-1171.

[21] 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 Naturelles, ATALA et AFCP, Nancy, France, 2020, pp. 108–121. URL: https://aclanthology.org/2020.jeptalnrecital-taln.9.

[22] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. doi:10.18653/v1/2020.emnlp-demos.6.

Theoretical study and empirical investigation of sentence analogies

Stergos Afantenos[1], Suryani Lim[2], Henri Prade[1] and Gilles Richard[1]

[1] *IRIT, University of Toulouse, France*

[2] *Federation University - Churchill -Australia*

Abstract

Analogies between 4 sentences, "a is to b as c is to d", are usually defined between two pairs of sentences (a, b) and (c, d) by constraining a relation R holding between the sentences of the first pair, to hold for the second pair. From a theoretical perspective, three postulates define an analogy - one of which is the "central permutation" postulate which allows the permutation of central elements b and c. This postulate is no longer appropriate in sentence analogies since the existence of R offers no guarantee in general for the existence of some relation S such that S also holds for the pairs (a, c) and (b, d). In this paper, the "central permutation" postulate is replaced by a weaker "internal reversal" postulate to provide an appropriate definition of sentence analogies. To empirically validate the aforementioned postulate, we build a LSTM as well as baseline Random Forest models capable of learning analogies based on quadruplets. We use the Penn Discourse Treebank (PDTB), the Stanford Natural Language Inference (SNLI) and the Microsoft Research Paraphrase (MSRP) corpora. Our experiments show that our models trained on samples of analogies between (a, b) and (c, d), recognize analogies between (b, a) and (d, c) when the underlying relation is symmetrical, validating thus the formal model of sentence analogies using "internal reversal" postulate.

1. Introduction

Analogy plays a crucial role in human cognition and intelligence. It has been characterized as "the core of cognition" [1] and has recently gained some interest from the computational linguistics and machine learning communities (see [2, 3]). Word analogies[1] such as "Paris is to France as Berlin is to Germany" are now well captured via word embeddings [4, 5]. If $\vec{a}, \vec{b}, \vec{c}, \vec{d}$ are the embeddings of words a, b, c, d, then $a : b :: c : d$[2] holds iff $(\vec{a}, \vec{b}, \vec{c}, \vec{d})$ is a parallelogram in the underlying vector space [6].

Although analogies between words have been extensively studied, analogies between sentences have received very scant attention by the community, to the best of our knowledge. Instead of dealing with words, dealing with sentences leads to 2 challenges:

- How to embed sentences in a vector space?

IARML@IJCAI-ECAI'2022: Workshop on the Interactions between Analogical Reasoning and Machine Learning, at IJCAI-ECAI'2022, July, 2022, Vienna, Austria

*Corresponding author.

[1] In the following, 'analogy' refers to a quaternary relation linking 4 items of the form "a is to b as c is to d", called analogical proportion.

[2] $a : b :: c : d$ is a standard notation for analogical proportion.

- How do we define a sentence analogy?

We expect sentence embeddings to be dense vectors supposed to reflect semantic properties of a sentence. Various approaches are available: embed each word and get the average of the vectors. In that case, the order of the words is lost. Another option, described in [7], allowing to recover the sentence from its embedding, makes use of Discrete Cosine Transform.

The question of defining sentence analogy is especially delicate. Indeed the aforementioned parallelogram model used for words reflects the usual postulates of analogies, namely if $a : b :: c : d$ holds, $c : d :: a : b$ (symmetry) and $a : c :: b : d$ (central permutation) should hold as well. This latter postulate (already questionable between words [8]) is still more debatable with sentences. In the NLP community, analogies between sentences are usually induced from predefined relationships between sentences. A quadruplet of sentences a, b, c, d defines an analogy $a : b :: c : d$ if the (implicit or explicit) relation that holds between the sentences of the first pair (a, b), also holds for the second pair (c, d). Let us consider the following example:

John sneezed loudly (a). Mary was startled (b).
Bob took an analgesic (c). His headache stopped (d).

In that case, the implicit relation R between sentences in a pair is a kind of causal relation. This example indicates that central permutation makes no sense here and raises the question of defining a weaker notion of analogy obeying another system of postulates. By which postulate to replace the central permutation? In this paper, we propose to introduce a postulate we call "internal reversal" that expresses that if $a : b :: c : d$ holds then $b : a :: d : c$ holds as well, and we study its consequences. So our main goal is to:

- theoretically investigate the formal consequences of this new model,
- empirically validate the model by implementing various classifiers of sentence analogies.

After presenting in Section 3 the standard formal definitions of analogies, including the "central permutation" postulate, and their immediate consequences, we focus on the replacement of the "central permutation" postulate by the *internal reversal* postulate. Having a better fit with what is accepted as sentence analogies in the NLP community, this postulate also impacts the machine learning perspective that we implement.

For natural language sentences, "internal reversal", as a formal postulate, may have some limitations. For instance, if $R = R^{-1}$, where R is the common relation that holds between two pairs of sentences (a, b) and (c, d) (e.g. , a is a *paraphrase* of b), one would expect that internal reversal holds straightforwardly. In that case, a machine learning model trained to recognize $a : b :: c : d$, should also recognize $b : a :: d : c$.

We investigate the conditions under which a machine learning model containing quadruplets of sentences (a, b, c, d) representing positive and negatives instances of analogies, is capable of identifying analogies for which the operation of *internal reversal* has been performed. We have then devised several series of experiments using various underlying models and datasets. The paper is structured as follows. After reviewing the related work (Section 2), in Section 3 we recall the formal definitions of analogical proportions and investigate the new case of sentence analogies, suggesting "internal reversal" postulate as a better fit and examining its consequences. In Section 4, we consider the consequences of the formal definition in a machine learning perspective, by suggesting a rigorous extension of an initial training set. Sections 5 and

6 are dedicated to the description to the context, protocol and results of our experiments. This work is an extension of [9], replacing artificially created datasets with human annotated ones.

2. Related work

Due to the advent of neural models and distributed representations of words, lexical analogies have been the focus of various works in computational linguistics. [10, 11, 12, 13, for example]. In terms of analogies on the sentential level few works exist. [14] investigate how existing embedding approaches can capture sentential analogies. They create two different kinds of datasets one consisting of replacing words with word analogies from the Google word analogy dataset [15] while the other is based on analogies between sentences that share common relations (entailment, negation, passivization, for example) or syntactic patterns (comparisons, opposites, plurals among others). The goal is to optimize $\arg max_{d \in V}(\vec{v_d}, \vec{v_b} - \vec{v_a} + \vec{v_c})$ with the additional constraint that $d \notin \{a, b, c\}$. Using these datasets, analogies are evaluated using various embeddings, such as GloVe [5], word2vec [15], fastText [16, 17], etc. showing that capturing syntactic analogies based on lexical analogies from the Google word analogies dataset is more effective than recognising analogies based on more semantic information. [18] use a similar approach to identify the most plausible answer a_i to a given question q from a pool A of answers to a question by leveraging analogies between (q, a_i) and various pairs of what they call *"prototypical"* question/answer pairs, assuming that there is an analogy between (q, a_i) and the prototypical pair (q_p, a_p). The goal is to select the candidate answer $a_i^* \in A$ such that:

$$a_i^* = \arg min_i(||(q_p - a_p) - (q - a_i)||)$$

. The authors limit the question/answer pairs to $wh-$ questions from WikiQA and TrecQA. They use a Siamese bi-GRUs as their architecture to represent the four sentences. In this manner, the authors learn embedding representations for the sentences which they compare against various baselines including random vectors, *word2vec, InferSent* and *Sent2Vec* obtaining better results with the WikiQA corpus. Most of the tested sentence embedding models succeed in recognizing syntactic analogies based on lexical ones but had a harder time capturing analogies between pairs of sentences based on semantics.

Instead of training a model to select the best candidate amongst a given set of candidates ([18, 19] train an encoder-decoder model based on LSTMs to generate the d given a pair (a, b) and a candidate c. Authors obtain vector encodings of $\vec{a}, \vec{b}, \vec{c}$ using an LSTM guided by two loss functions. The authors then experiment with concatenation, summation and arithmetic analogy on these vectors to obtain a new vector which is then used as input for the decoding mechanism, showing that arithmetic analogy outperforms the other methods.

In this paper, the aim is to empirically validate the "internal reversal" postulate (without focusing on accuracy). To our knowledge, such a study has not been conducted before.

3. Theoretical Foundations of Analogies

We briefly recall the formal definition of analogy such as found in [20, 21, 22]. We focus on a widely accepted definition for sentence analogies and we investigate to what extent sentence analogies obey the formal postulates and what has to be modified in the formal setting to fit with this particular definition.

3.1. Formal definitions

Given a set of items X, a (proportional) analogy is a quaternary relation supposed to obey the 3 following postulates (e.g.,[21]):

$\forall a, b, c, d \in X$:

1. $a : b :: a : b$ (*reflexivity*);
2. $a : b :: c : d \to c : d :: a : b$ (*symmetry*);
3. $a : b :: c : d \to a : c :: b : d$ (*central permutation*).

These postulates have straightforward consequences like:

- $a : a :: b : b$ (*identity*);
- $a : b :: c : d \to b : a :: d : c$ (*internal reversal*);
- $a : b :: c : d \to d : b :: c : a$ (*extreme permutation*);
- $a : b :: c : d \to d : c :: b : a$ (*complete reversal*).

Among the 24 permutations of a, b, c, d, the previous postulates induce 3 distinct classes each containing 8 distinct proportions regarded as equivalent due to postulates: $a : b :: c : d$ has in its class $c : d :: a : b$, $c : a :: d : b$, $d : b :: c : a$, $d : c :: b : a$, $b : a :: d : c$, $b : d :: a : c$, and $a : c :: b : d$. But $b : a :: c : d$ and $a : d :: c : b$ do not belong to the class of $a : b :: c : d$ and are elements of the two other classes.

3.2. Sentence analogies

In the NLP community, the 4 items a, b, c, d are sentences in natural language, not necessarily the same. It is widely admitted that the sentences are in analogy (i.e., $a : b :: c : d$) as soon as there is a relation R, the relation between sentences, such that $R(a, b)$ and $R(c, d)$. The example from the introduction is a perfect illustration of this definition where the relation R is just causality:

John sneezed loudly (a). Mary was startled (b).
Bob took an analgesic (c). His headache stopped (d).

But:

Il fait beau aujourd'hui (a). Today we have nice weather (b).
Il vaut mieux eviter la guerre (c). It is better to avoid war (d).

is another example of analogies between sentences where the implicit relation R is "b is the English translation of the French sentence a". From a logical viewpoint, this can be expressed as:

$$a : b :: c : d \text{ iff } \exists R \text{ s.t. } R(a, b) \wedge R(c, d) \tag{1}$$

where $\wedge$ is just the formal notation for the and connector. This definition can be considered as quite vague because, as advocated in [23, 24], there is always a way to find such a relation

R between 2 sentences. A more effective option used in the NLP community is to consider that the underlying relation R belongs to a finite set S of relations. Such relations can be, for example, discourse relations (*Elaboration, Continuation, Contrast, Concession*, etc) or a Causality relation as is the case in the above example. Then, the formal definition has to be refined into:

$$a : b :: c : d \text{ iff } \exists R \in S \text{ s.t. } R(a, b) \wedge R(c, d) \tag{2}$$

where $S = \{R_1, \ldots, R_n\}$ is just a finite non-empty set of relations belonging to a list of target relations. With this definition, we constraint the relation R to belong to a predefined set. Obviously, in the case of French-English translation, the list S is reduced to only one relation. It is quite clear that reflexivity and symmetry are still valid postulates for sentence analogies i.e., they are satisfied with both above definitions. Back to our initial example:

John sneezed loudly (a). Mary was startled (b).
Bob took an analgesic (c). His headache stopped (d).

Definition 1 or 2 still applies to $c : d :: a : b$:

Bob took an analgesic (c). His headache stopped (d).
John sneezed loudly (a). Mary was startled (b).

which is then a valid analogy. Nevertheless, central permutation is not satisfied with the above definitions 1 or 2.

3.3. Internal reversal for sentence analogies

Let us now focus on the "internal reversal" postulate as a alternative to "central permutation":

$a : b :: c : d \to b : a :: d : c$ (internal reversal)

By definition, if $R(a, b)$ holds then $R^{-1}(b, a)$ holds. Definition 1 supports "internal reversal": for instance, if relation $R(a, b)$ is interpreted as "a is a cause of b", $R^{-1}(b, a)$ can be the passive form "b is a consequence of a". But Definition 2 does not support "internal reversal" except if, for each relation R in the set S of built-in relations, we also have its counterpart R^{-1}. A simple way to ensure this property is to consider relations R such that $R = R^{-1}$. For instance, $R(a, b)$ is defined as "a is a paraphrase of b".

In the general case, a proper definition of a sentence analogy supporting the 3 postulates (reflexivity, symmetry, internal reversal) would be:

$$\begin{array}{l} a : b :: c : d \text{ iff } \exists R \in S \text{ s.t.} \\ \begin{cases} (R(a, b) \wedge R(c, d)) \\ \vee (R^{-1}(a, b) \wedge R^{-1}(c, d)) \end{cases} \end{array} \tag{3}$$

This leads to a formal definition of sentence analogies with:

1. $a : b :: a : b$ (*reflexivity*);
2. $a : b :: c : d \to c : d :: a : b$ (*symmetry*);
3. $a : b :: c : d \to b : a :: d : c$ (*internal reversal*).

As immediate consequences, we get that :

- there are only 4 equivalent forms (instead of 8 with the central permutation postulate) for an analogy:

- $a : b :: c : d$, $c : d :: a : b$, $d : c :: b : a$, and $b : a :: d : c$.
- $a : b :: c : d \rightarrow d : c :: b : a$ (*complete reversal*).
- $a : a :: a : a$ (full identity) is still satisfied.
- $a : a :: b : b$ (identity) is no longer a consequence of the new postulates.

4. Implications for Machine Learning

Let us assume that we have at our disposal a repository of pairs of sentences (a, b) with their associated relation R. From this repository, we need a training set of examples for the classifier. Given the previous section, several steps can be implemented.

1) Building an initial training set of analogies $a : b :: c : d$ can be done by joining 2 pairs (a, b) and (c, d) belonging to the same relation R. This constitutes a set of positive examples $\mathcal{X}^+$ such that for every quadruplet $(a, b, c, d) = \mathbf{x} \in \mathcal{X}^+$ the training instances are $\{\mathbf{x}, y\}$ with $y = 1$. In terms of negative examples, joining 2 pairs (a, b) and (c, d) belonging to different relations leads to build a set of negative examples $\mathcal{X}^-$ such that for every quadruplet $(a, b, c, d) = \mathbf{x} \in \mathcal{X}^-$ the training instances are $\{\mathbf{x}, y\}$ with $y = 0$. The training set $\mathcal{X} = \mathcal{X}^+ \cup \mathcal{X}^-$ is then a set of quadruplets of sentences a, b, c, d such that:

- if the implicit/explicit relation R between the pair (a, b) also holds for the pair (c, d), then $(a, b, c, d) \in \mathcal{X}^+$
- if the implicit/explicit relation R between the pair (a, b) does not hold for the pair (c, d), then $(a, b, c, d) \in \mathcal{X}^-$

Applying symmetry postulate allows to double the size of $\mathcal{X}^+$, just by adding $(c, d, a, b) \in \mathcal{X}^+$ as soon as $(a, b, c, d) \in \mathcal{X}^+$. We then improve the theoretical unbalance between $\mathcal{X}^+$ and $\mathcal{X}^-$.

2) The same method applies with internal reversal postulate, by adding $(b, a, d, c) \in \mathcal{X}^+$ as soon as $(a, b, c, d) \in \mathcal{X}^+$. This again doubles the size of $\mathcal{X}^+$.

At this stage, we have multiplied by 4 the initial size of our positive training set $\mathcal{X}^+$ by introducing common sense analogies deducible from the initial ones, but not necessarily related to the initial list of relations S. Can we do more?

The Identity Relation For completeness sake, one could argue that it is still possible to extend the set of positive examples since it seems acceptable to consider $a : a :: b : b$ as a valid sentence analogy even though identity Id relation likely does not belong to S. But identity relation Id holds between the pairs (a, a) and (b, b). Although recognition of analogies based on the identity relation might seem trivial from an NLP perspective it could still be a useful task in case that we want to evaluate the quality of our classifier. In other words, if a potential classifier is not able to identify analogies based on the identity relation, one should probably reconsider the underlying approach.

The Inverse Relation A scenario that appears quite often in Natural Language Processing, although far from being a generalized phenomenon, is that a relation R between sentences (or larger proportions of text for that matter) is its own inverse R^{-1}.

Instances of such a relation can be, for example, that of the *paraphrase*. If a is a paraphrase of b, obviously b is a paraphrase of a. The same hold for the operation of translation. If sentence a is a translation of b then again b is a translation of a. Following our initial definition of analogy, we will have to accept:

$a : b :: b : a$ when R is its own inverse

Before moving to the details of the empirical validation, we describe the datasets we use in the following section.

5. Experiments

As explained earlier in this paper, our main goal is the empirical validation of internal reversal for sentential analogies, using various corpora. To investigate this postulate we devise the following sets of experiments.

5.1. Experimental settings

Base setting Given a training set

$$(\mathcal{X}_{train}, \mathcal{Y}_{train}) = (\{\mathbf{x}_i\}_{i=1}^{n}, \{y_i\}_{i=1}^{n}\})$$

and a test set $(\mathcal{X}_{test}, \mathcal{Y}_{test}) = (\{\mathbf{x}_i\}_{i=1}^{m}, \{y_i\}_{i=1}^{m})$ with m typically being a tenth of n and $\mathbf{x}_j$ representing a quadruplet of sentences $a : b :: c : d$[3] and $y_i \in \{0, 1\}$ we learn a model $\mathcal{H}_b$ capable of identifying analogies with a certain accuracy. Crucially, $|\{y_k : y_k = 1\}| = |\{y_k : y_k = 0\}|$ both for training and testing sets. Due to the huge size of instances at our disposal, there is no need at this stage to implement any further data augmentation process, as explained in the previous Section. In other words, we have an equal number of positive and negative instances in training and testing sets, for a total of 4M instances.

Internal reversal on the test set (Experimental setting 1) In this series of experiments, we used the same training set $(\mathcal{X}_{train}, \mathcal{Y}_{train}) = (\{\mathbf{x}_i\}_{i=1}^{n}, \{y_i\}_{i=1}^{n}\})$ as the base setting $\mathcal{H}_b$. To construct the test set, we perform *internal reversal* on all the instances of the train set that we have used in base setting. Our goal is to see whether we get similar results on analogies for the internal reversal.

Test set from train distribution with internal reversal (Experimental setting 2) For this series of experiments we use the same training set $(\mathcal{X}_{train}, \mathcal{Y}_{train}) = (\{\mathbf{x}_i\}_{i=1}^{n}, \{y_i\}_{i=1}^{n}\})$ for the base setting $\mathcal{H}_b$. The test set though is constructed in the following way: for every positive instance $(\mathbf{x}_{a:b::c:d}, 1)$ in $(\mathcal{X}_{train}, \mathcal{Y}_{train})$ we add the *internal reversal* pair $(\mathbf{x}_{b:a::d:c}, 1)$ to the new testing set $(\mathcal{X}_{test}, \mathcal{Y}_{test})$ whose size thus is $n/2$. In contrast to experimental setting 1 where the underlying sentences between train and test distributions are different, in this series of experiments we want to see how well a trained model can detect analogies after performing internal reversal on the same set of pairs of sentences.

[3] Henceforth, we will denote a representation for a quadruplet of sentences $a : b :: c : d$ by the vector $\mathbf{x}_{a:b::c:d}$.

Augmenting training and test sets (Experimental setting 3) In the series of experiments we learn a model $\mathcal{H}_a$ using

$$(\mathcal{X}^a_{train}, \mathcal{Y}^a_{train}) = (\{\mathbf{x}_i\}_{i=1}^{n+n/2}, \{y_i\}_{i=1}^{n+n/2}\})$$

and a test set

$$(\mathcal{X}^a_{test}, \mathcal{Y}^a_{test}) = (\{\mathbf{x}_i\}_{i=1}^{m+m/2}, \{y_i\}_{i=1}^{m+m/2})$$

where both train and test sets have been augmented using the following rule: for each instance $(\mathbf{x}_{a:b::c:d}, 1)$ in train or test set we add the following instance $(\mathbf{x}_{b:a::d:c}, 1)$. In other words, we double *only* the positive instances by adding the internal reversal of a quadruplet as a positive instance.

Augmenting test set (Experimental setting 4) In this series of experiments the train set and thus the model learnt is the same as the base setting. In other words, we have a training set $(\mathcal{X}_{train}, \mathcal{Y}_{train}) = (\{\mathbf{x}_i\}_{i=1}^n, \{y_i\}_{i=1}^n\})$ from which we learn a model $\mathcal{H}_b$. For testing though we have a new test set $(\mathcal{X}^{a_t}_{test}, \mathcal{Y}^{a_t}_{test}) = (\{\mathbf{x}_i\}_{i=1}^m, \{y_i\}_{i=1}^m)$ which results from $(\mathcal{X}_{test}, \mathcal{Y}_{test})$ of the base setting by keeping only positive instances. This subset is then augmented with instances $(\mathbf{x}_{b:a::d:c}, 1)$ for every instance $(\mathbf{x}_{a:b::c:d}, 1)$ we have in that subset, resulting thus in m total positive instances.

5.2. Datasets

To perform our experiments, we used three corpora: Penn Discourse TreeBank (PDTB), Stanford Natural Language Inference Corpus (SNLI) and the paraphrase dataset MPRC.

PDTB dataset The first dataset that we use is PDTB version 2.1[25]. (36,000 pairs of sentences annotated with discourse relations). Relations can be explicitly expressed via a discourse marker, or implicitly expressed in which case no such discourse marker exists and the annotators provide one that more closely describes the implicit discourse relation. Relations are organized in a taxonomy of depth 3. Level 1 (L1) (top level) has four types of relations (*Temporal, Contingency, Expansion and Comparison*), level 2 (L2) has 16 relation types and level 3 (L3) has 23 relation types. For this series of experiments, we used the L1 relation.

SNLI dataset SNLI is a corpus of pairs of sentences from [26]. SNLI was created and annotated manually. It contains 570K human-written sentence pairs considered as a sufficient number of pairs for machine learning. The sentence pairs are annotated with entailment, contradiction and semantic independence. More precisely, a pair of sentences a and b can be annotated either with *Entailment, Contradiction* or *Neutral* relation. Construction of the corpus was done using Mechanical Turk who was presented with a premise in the form of a sentence and was asked to provide three hypotheses, in a sentential form, for each of the aforementioned labels. 10% of the corpus was validated by trusted Mechanical Turks. Overall a Fleiss κ of 0.70 was achieved. For our experiments we considered the Neutral relation as symmetric.

MRPC dataset The third corpus is Microsoft Research Paraphrase Corpus (MRPC [27]). It contains about 5800 pairs of sentences which can either be a paraphrase of each other or not. Each pair of sentences was annotated by two annotators. In case of disagreements, a third annotator resolved the conflict. After this, about two-thirds of the pairs were annotated as paraphrases and one third as not.

5.3. Embedding techniques

There are well-known word embeddings such as word2vec [15], Glove [5], BERT [28], fastText [17], etc. It is standard to start from a word embedding to build a sentence embedding. Sentence embedding techniques represent entire sentences and their semantic information as vectors. In this paper, we focus on 2 techniques relying on initial word embedding.

- The simplest method is to average the word embeddings of all words in a sentence. Although this method ignores both the order of the words and the structure of the sentence, it performs well in many tasks. So the final vector has the dimension of the initial word embedding.

- The other approach, suggested in [7], makes use of the Discrete Cosine Transform (DCT) as a simple and efficient way to model both word order and structure in sentences while maintaining practical efficiency. Using the inverse transformation, the original word sequence can be reconstructed. A parameter l is a small constant that needs to be set. One can choose how many features are being embedded per sentence by adjusting the value of l, but undeniably it increases the final size of the sentence vector by a factor l. If the initial embedding of words is of dimension n, the final sentence dimension will be $= n * l$ (see [7] for complete description). In our experiments, we use the average method to embed sentences as it is at least as effective as DCT [9].

5.4. Models

Random Forest (RF) We have tested our hypothesis on a classical method successfully used for word analogy classification [29]: Random Forests (RF). The parameters for RF are 100 trees, no maximum depth, and a minimum split of 2. We also use LSTMs, but any other model (SVM, etc.), could have been used.

Bi-LSTM architecture Given a quadruplet of sentences $a : b :: c : d$ which can be an analogy or not, we represent each sentence by its input tokens $a = \{w_1^a, \ldots, w_k^a\}, b = \{w_1^b, \ldots, w_k^b\}, c = \{w_1^c, \ldots, w_k^c\}$ and $d = \{w_1^d, \ldots, w_k^d\}$. Although sentences can have different lengths we have empirically fixed $k = 35$; if a sentence has less than 35 word tokens we use padding. Each word token w_i^s (with $s \in \{a, b, c, d\}$ and $i \in [1 \ldots k]$) is represented by a Glove vector of 300 dimensions. In this series of experiments, LSTM did not use averaging or DCT since the recurrent nature of LSTMs themselves accounts for the structure of a sentence. Our architecture is composed by *four* Bi-LSTMs whose output is passed over to a feed-forward network. More precisely, for each sentence we recursively calculate $h_t = o_t \otimes tanh(C_t)$ with $\otimes$ representing the Hadamard operation and

$$o_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o)$$

where

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{\mathbf{C}}_t$$

and

$$i_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i)$$

$$\tilde{\mathbf{C}}_t = tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C)$$

$$f_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f)$$

In the above, $\mathbf{x}_t$ represents the vector for token w_t in a given sentence. These representations are obtained on both directions. Thus for each sentence the following representations are obtained:

$$a = \{w_i^a\} = \overrightarrow{h_t^a} \mathbin{+\!\!+} \overleftarrow{h_t^a}; b = \{w_i^b\} = \overrightarrow{h_t^b} \mathbin{+\!\!+} \overleftarrow{h_t^b}; c = \{w_i^c\} = \overrightarrow{h_t^c} \mathbin{+\!\!+} \overleftarrow{h_t^c}; d = \{w_i^d\} = \overrightarrow{h_t^d} \mathbin{+\!\!+} \overleftarrow{h_t^d}$$

with $\mathbin{+\!\!+}$ representing the concatenation operation.

The above representations are given as input to a single layer feed forward network:

$$\mathbf{h}_f = f(\mathbf{W}^T \mathbf{h}_{LSTM} + \mathbf{b})$$

with

$$\mathbf{h}_{LSTM} = \overrightarrow{h_t^a} \mathbin{+\!\!+} \overleftarrow{h_t^a} \mathbin{+\!\!+} \overrightarrow{h_t^b} \mathbin{+\!\!+} \overleftarrow{h_t^b} \mathbin{+\!\!+} \overrightarrow{h_t^c} \mathbin{+\!\!+} \overleftarrow{h_t^c} \mathbin{+\!\!+} \overrightarrow{h_t^d} \mathbin{+\!\!+} \overleftarrow{h_t^d}$$

using Rectified Linear Unit (ReLU) as activation function. Finally, the prediction is performed using a sigmoïd function:

$$\hat{y} = \sigma(\mathbf{W}^T \mathbf{h}_{LSTM} + \mathbf{b}) = \frac{1}{1 + e^{-\mathbf{W}^T \mathbf{h}_{LSTM} + \mathbf{b}}}$$

The architecture is guided by a standard binary cross entropy loss function.

6. Results and discussion

Results of our experiments for LSTMs and RFs are shown in Tables 1 and 2 respectively. In all cases, we randomly generated quadruplets (a, b, c, d) which we annotated as analogies (class 1) if pairs (a, b) and (c, d) shared the same relation, or with class 0 if they did not. For PDTB and SNLI we randomly generated 2 million instances for training; testing and development corpus contained 200.000 instances each. For the paraphrase corpus, we generated 4 million instances for training and testing and development corpus contained 200.000 instances each. Each dataset contains an equal number of positive and negative instances. As we can see, base settings for all datasets perform quite moderately, which is to be expected since our aim was not to create a general model for sentential analogies, which would require much more data and powerful models with billions of parameters. Instead, our goal was to examine under which conditions internal reversal holds. As we can see, in experimental setting 1, for which the test set is the same as the train but with internal reversal, results on PDTB and SNLI, which contain relations that are not symmetric, are worse than the base setting. This is not the case though for the paraphrases corpus for which results are better to the base setting.

In the second set of experiments, we decided to focus solely on the positive instances and examine if learning analogy $a : b :: c : d$ also implicitly learnt internal reversal, that is $b : a :: d : c$. We used the same base setting that we have learnt, but for testing, we created a new dataset. Starting from an empty set, we took every positive instance of the training set and performed an internal reversal; we then add it to the new test dataset. The resulting dataset has no common instances with the training dataset, but every instance of it is an internal reversal of the positive instances of the training set. As we can see there is almost no difference in scores for PDTB and SNLI, but the results for the paraphrases corpus (93.412% F_1 for LSTMs and 87.544% for RFs) clearly show that when a relation is *symmetrical* the model almost makes no difference between an analogy and its internal reversal. It is interesting to observe that the trend for LSTM is similar to RF. However, the results from LSTM appear to be more stable. In the third series of experiments, we augmented both the training and testing datasets with the internal reversal. All three datasets showed a significant increase—of almost 20 percentile points for some cases—for the detection of analogies. In the fourth and final set of experiments, we used the same base setting that we had used initially. The test set was constructed based on the same test set as the base setting but we removed all negative instances and focused solely on the positive ones augmented with the internal reversal. Again here we can see a significant increase in the results for the detection of analogies, further showing that the model learns internal reversal as well. On Table 1 , Experiment setting 2 for MRPC has the highest F1: this corpus has more symmetrical relationships when compared to PDTB and SNLI. We observed with Table 2 the trend already observed with LSTM: Experiment setting 2 has the highest F1.

		Precision	Recall	F1	Accuracy
	PDTB				
base setting	class 1	54.274	47.476	50.648	53.739
	class 0	53.322	60.001	56.465	
Exp. Set. 1	class 1	48.91	39.76	43.863	49.114
	class 0	49.254	58.468	53.467	
Exp. Set. 2	class 1	100.0	39.76	56.898	39.76
Exp. Set. 3	class 1	70.16	79.346	74.471	63.733
	class 0	44.038	32.507	37.404	
Exp. Set. 4	class 1	100.0	46.585	63.56	46.585
	SNLI				
base setting	class 1	67.862	67.811	67.837	67.859
	class 0	67.856	67.907	67.882	
Exp. Set. 1	class 1	50.111	49.57	49.839	50.11
	class 0	50.11	50.651	50.379	
Exp. Set 2	class 1	100.0	49.57	66.283	49.57
Exp. Set 3	class 1	84.489	83.982	84.235	79.047
	class 0	68.365	69.185	68.772	
Exp. Set. 4	class 1	100.0	59.086	74.282	59.086
	MRPC				
base setting	class 1	53.45	61.487	57.188	53.969
	class 0	54.671	46.45	50.227	
Exp. Set. 1	class 1	80.454	87.638	83.892	83.173
	class 0	86.426	78.708	82.387	
Exp. Set. 2	class 1	100.0	87.638	**93.412**	87.638
Exp. Set. 3	class 1	69.033	72.395	70.674	59.946
	class 0	38.832	35.05	36.844	
Exp. Set. 4	class 1	100.0	62.752	77.114	62.75

Table 1
Results for LSTM

		Precision	Recall	F1	Accuracy
	PDTB				
base setting	class 1	54.604	33.778	41.737	53.314
	class 0	52.744	72.468	61.053	
Exp. Set. 1	class 1	51.254	31.096	38.708	50.826
	class 0	50.639	70.504	58.943	
Exp. Set. 2	class 1	100.00	31.096	47.440	31.096
Exp. Set. 3	class 1	66.263	99.953	79.694	66.267
	class 0	69.847	0.213	0.424	
Exp. Set. 4	class 1	100.00	32.117	48.619	32.117
	SNLI				
base setting	class 1	50.725	47.006	48.794	50.729
	class 0	50.732	54.443	52.522	
Exp. Set. 1	class 1	50.302	46.189	48.158	50.285
	class 0	50.270	54.379	52.244	
Exp. Set 2	class 1	100.00	46.189	63.191	46.189
Exp. Set 3	class 1	70.368	86.903	77.766	66.898
	class 0	50.797	26.979	35.241	
Exp. Set. 4	class 1	100.00	46.313	63.307	46.313
	MRPC				
base setting	class 1	54.327	69.353	60.927	54.739
	class 0	55.502	39.599	46.221	
Exp. Set. 1	class 1	58.916	77.847	67.071	61.374
	class 0	66.313	44.547	53.293	
Exp. Set. 2	class 1	100.00	77.847	**87.544**	77.847
Exp. Set. 3	class 1	67.523	99.649	80.499	67.437
	class 0	48.952	0.698	1.377	
Exp. Set. 4	class 1	100.0	69.139	81.754	69.139

Table 2
Results for Random Forest

7. Conclusion and future work

In this paper, we have suggested a new formal model dedicated to sentence analogies, replacing the standard model for word analogies. A weaker “internal reversal” postulate takes the place of the well-known “central permutation” postulate. From a purely formal viewpoint, we have investigated the consequences of this new model and to what extent it fits with sentence analogies. To validate this approach in practice, we have implemented sentence analogies classifiers, using well-known machine learning algorithms. We have also designed two machine learning protocols involving different ways to build a training set, all derived from the formal expected properties. Our results show that an “internal reversal” sentence analogy is recognized by our algorithms as a valid analogy as soon as the underlying relation between sentences is symmetric (e.g. “to be a paraphrase of”). When this relation is not symmetric (e.g., “to be a consequence of”), “internal reversal” sentence analogies are not always recognized. Maybe, in the general case, learning R is not the same as learning R^{-1}. Alternatively finding a more accurate postulate might be a valid track of research for the future. Analogy postulates could also be used for further constraining the classifier.

Acknowledgments

The authors would like to express their gratitude to the anonymous reviewers for their valuable comments. They would also like to thank the organizers of this workshop.

References

[1] D. R. Hofstadter, Analogy as the Core of Cognition, MIT Press, 2001, pp. 499–538.

[2] C. Allen, T. Hospedales, Analogies explained: Towards understanding word embeddings, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 223–231. URL: http://proceedings.mlr.press/v97/allen19a.html.

[3] F. Chollet, On the measure of intelligence, 2019. arXiv:1911.01547.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: C. J. C. B. et al. (Ed.), Advances in Neural Information Processing Systems 26, Curran Associates Inc., 2013, pp. 3111–3119.

[5] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: EMNLP, 2014, pp. 1532–1543.

[6] D. E. Rumelhart, A. A. Abrahamson, A model for analogical reasoning, Cognitive Psychol. 5 (1973) 1–28.

[7] N. Almarwani, H. Aldarmaki, M. Diab, Efficient sentence embedding using discrete cosine transform, in: EMNLP, 2019, pp. 3663–3669.

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

[9] S. Afantenos, T. Kunza, S. Lim, H. Prade, G. Richard, Analogies between sen-

tences:theoretical aspects - preliminary experiments, in: Proc. 16th Europ. Conf. Symb. & Quantit. Appr. to Reas. with Uncert. (ECSQARU), 2021.

[10] Z. Bouraoui, S. Jameel, S. Schockaert, Relation induction in word embeddings revisited, in: COLING, 1627-1637, Assoc. Computat. Ling., 2018.

[11] A. Drozd, A. Gladkova, S. Matsuoka, Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen, in: COLING, 2016, pp. 3519–3530.

[12] P. D. Turney, A uniform approach to analogies, synonyms, antonyms, and associations, in: COLING, 2008, pp. 905–912.

[13] P. D. Turney, Distributional semantics beyond words: Supervised learning of analogy and paraphrase, TACL 1 (2013) 353–366.

[14] X. Zhu, G. de Melo, Sentence analogies: Linguistic regularities in sentence embeddings, in: COLING, 2020.

[15] T. Mikolov, K. Chen, G. S. Corrado, J. Dean, Efficient estimation of word representations in vector space, CoRR abs/1301.3781 (2013).

[16] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, in: Transactions of the Association for Computational Linguistics, 2017, p. 135–146.

[17] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, A. Joulin, Advances in pre-training distributed word representations, in: Proc. of LREC, 2018.

[18] A. Diallo, M. Zopf, J. Fürnkranz, Learning analogy-preserving sentence embeddings for answer selection, in: Proc. 23rd Conf. Computational Natural Language Learning, 910 - 919, Assoc. Computat. Ling., 2019.

[19] L. Wang, Y. Lepage, Vector-to-sequence models for sentence analogies, in: 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2020, pp. 441–446. doi:10.1109/ICACSIS51025.2020.9263191.

[20] Y. Lepage, De l'analogie rendant compte de la commutation en linguistique, Habilit. à Diriger des Recher., Univ. J. Fourier, Grenoble (2003). URL: https://tel.archives-ouvertes.fr/tel-00004372/en.

[21] Y. Lepage, Analogy and formal languages, Electr. Notes Theor. Comput. Sci. 53 (2001).

[22] H. Prade, G. Richard, From analogical proportion to logical proportions, Logica Univers. 7 (2013) 441–505.

[23] M. Hesse, On defining analogy, Proceedings of the Aristotelian Society 60 (1959) 79–100.

[24] M. Hesse, Analogy and confirmation theory, Philosophy of Science xxxi (1964) 319–327.

[25] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, B. Webber, The Penn Discourse TreeBank 2.0., in: LREC 08, 2008. URL: http://www.lrec-conf.org/proceedings/lrec2008/pdf/754_paper.pdf.

[26] S. R. Bowman, G. Angeli, C. Potts, C. D. Manning, A large annotated corpus for learning natural language inference, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2015.

[27] W. B. Dolan, C. Brockett, Automatically constructing a corpus of sentential paraphrases, in: Proceedings of the Third International Workshop on Paraphrasing (IWP2005), 2005. URL: https://www.aclweb.org/anthology/I05-5002.

[28] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional

transformers for language understanding, CoRR abs/1810.04805 (2018).

[29] S. Lim, H. Prade, G. Richard, Solving word analogies: A machine learning perspective, in: Proc. 15th Europ. Conf. Symb. & Quantit. Appr. to Reas. with Uncert. (ECSQARU), LNCS 11726, 238-250, Springer, 2019.

Solving Morphological Analogies Through Generation

Kevin Chan[1,†], Shane P. Kaszefski-Yaschuk[1,†], Camille Saran[1,†], Esteban Marquer[1] and Miguel Couceiro[1,*]

[1] *Université de Lorraine, CNRS, LORIA, F-54000, France*

Abstract

This contribution is a first attempt at solving morphological analogies through generation, instead of relying on retrieval approaches. Our preliminary experiments show promising results for some languages and reveal the feasibility of the approach in generating solutions of analogical equations in the morphology setting.

Keywords

Morphological analogy, Analogy solving, Representation learning, Word generation

1. Introduction

Analogical proportions are understood as statements of the form "A is to B as C is to D" denoted $A : B :: C : D$, and they are the basis of analogical inference. Analogical inference is a remarkable capability of human reasoning, and that has been used to solve hard reasoning tasks. To some extent, it can be thought of as transferring knowledge from a source domain to a different, but somewhat similar, target domain by relying simultaneously on similarities and dissimilarities. Analogy based reasoning (AR) is closely related to case-based reasoning and has gained increasing interest from the artificial intelligence (AI) community, and has shown its potential in multiple machine learning (ML) tasks such as classification, decision making and recommendation with competitive results [1, 2, 3, 4]. Furthermore, analogical inference can support data augmentation through analogical extension and extrapolation for model learning, especially in environments with few labeled examples [5]. Also, it has been successfully applied to several classical NLP tasks such as machine translation [6], several semantic and morphological tasks [7, 8, 9], as well as (visual) question answering and solving puzzles and scholastic aptitude tests [10, 11].

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 analogical proportion. This task asks for a common theoretical framework. However, the notion of analogy is not

IARML@IJCAI-ECAI'2022: Workshop on the Interactions between Analogical Reasoning and Machine Learning, at IJCAI-ECAI'2022, July, 2022, Vienna, Austria

[*] Corresponding author.

[†] Equal contribution.

kevin.chan3@etu.univ-lorraine.fr (K. Chan); shane-peter.kaszefski-yaschuk5@etu.univ-lorraine.fr (S. P. Kaszefski-Yaschuk); camille.saran5@etu.univ-lorraine.fr (C. Saran); esteban.marquer@loria.fr (E. Marquer); miguel.couceiro@loria.fr (M. Couceiro)

0000-0003-2315-7732 (E. Marquer); 0000-0003-2316-7623 (M. Couceiro)

CEUR Workshop Proceedings (CEUR-WS.org)

consensual, and there have been several efforts that follow different axiomatic and logical approaches [12, 13]. For instance, [14] introduces the following 4 postulates in the linguistic context as a guideline for formal models of 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 $D = B$). Such postulates appear reasonable in the word domain, but they can be criticized in other application domains [15, 16].

The second basic task is *analogy solving* that refers to the task of extrapolating or generating, for a given triple A, B, C the value X such that $A : B :: C : X$ is a valid analogy. One approach to tackling this task is by retrieval and adaptation, *i.e.*, defining an X from a pool of retrieved candidate solutions to be suitably adapted. In fact, analogy solving is somewhat related to case-based reasoning (CBR) [17] where, given a set P of problems, a set S of solutions and a set $\mathcal{C}$ of cases $(x, y) \in P \times S$, the CBR task is to find a solution y_t to a given target problem x_t. CBR basically consists in (1) selecting k source cases in the case base according to some criteria related to the target problem (retrieval step), and (2) reusing the k retrieved cases for proposing a target solution (adaptation step). Despite being a reasonable approach in controlled settings, it suffers from several drawbacks: it requires a suitable choice of examples and is intrinsically limited by case based approaches, that prevent creative inference and innovation.

More recent approaches to analogy solving take advantage of recent deep neural network frameworks that rely on vector representations and on the structure of the underlying multidimensional space. Essentially, analogical proportions are formalized in terms of the *parallelogram rule* by which four vectors e_A, e_B, e_C, and e_D (representing four elements A, B, C, and D) are in analogical proportion if $e_D - e_C = e_B - e_A$. Such an arithmetic view of analogical proportions has been used since the first works on analogy [18], and it was the key element in the methodology employed by earlier neural-based approaches [19, 20]. In the absence of a decoder, the authors implicitly generate a representation e_X and then retrieve the closest candidate D from the vocabulary to solve the analogical equation $A : B :: C : X$ (see brief discussion of Subsection 2.2). However, Chen *et al.* [21] argue that the latter two methods significantly differ from human performance.

In the case of sentence analogies (*i.e.,* where A, B, C are sentences), [22] overcomes this issue by training a decoder that is then used to decode e_X. In this paper, we employ a similar approach in the setting of word analogies. More precisely, following the tracks of [23, 24, 25], we address morphological issues on words and tackle the problem of solving morphological analogies. Inspired by the work of [22] to solving sentence analogies, the novelty in our contribution is to make use of autoencoders to solving morphological analogies on words. More precisely, the main contributions of this paper are as follows: (*i*) we propose a model to generate words at character level from word embeddings with high reconstruction performance, and (*ii*) we achieve encouraging results to solving morphological analogies by generation, thus indicating the feasibility of the approach. Nonetheless, this constitutes ongoing research that requires further investigations.

The paper is organized as follows. We first briefly survey previous work on both main tasks dealing with morphological analogies in Section 2. We then describe the key components of the deep learning architecture as well as the analogy solving procedure we use in Section 3. The empirical setting setting is then presented in Section 4 where we also discuss the experimental results. We conclude with a general overview of this contribution in Section 5 and propose further directions of future research.

2. Related Approaches

In this paper, we focus on morphological analogies, *i.e.*, analogies on words A, B, C, and D that capture morphological transformations of words (*e.g.*, conjugation or declension). In this section we introduce key approaches of analogy detection and solving in morphology. The main trend follows the seminal work of [26] by exploiting the postulates of analogical proportions mentioned in introduction, but some approaches including ours take a slightly different approach. As deep learning approaches to morphological analogies are strongly related to approaches on semantic word analogies, the latter will also be discussed here.

2.1. Analogy Detection

As mentioned above, the analogy detection task corresponds to classifying quadruples A, B, C, D into valid or invalid analogies. The tools in [27] detect morphological analogies using the number of characters occurrences and the length of the longest common subword. Their approach is designed to generate analogical grids, *i.e.*, matrices of transformations of various words, similar to paradigm tables in linguistics [7]. A data-driven alternative was implemented by [8] for semantic word analogies. Using a dataset of semantic analogies, they learn a neural network to classify quadruples A, B, C, D into valid or invalid analogies, using their embedding e_A, e_B, e_C, and e_D. This approach was applied to morphological analogies in [24] by replacing the GloVe [28] semantic embeddings used by Lim et al. with a morphology-oriented word embedding model.

2.2. Analogy Solving

Approaches to analogy solving usually *generate* the fourth element to solve the analogy, but it is also possible to leverage a list of candidates and *retrieve* the most fitting fourth term to solve the analogy. In Subsubsection 2.2.2 we describe key approaches using the former method to solve morphological analogies, and similarly in Subsubsection 2.2.1 for the latter method. Many approaches in embedding spaces use the latter method because generation from an embedding space can be challenging, and we describe some in Subsubsection 2.2.1. However, such retrieval approaches are limited to the available vocabulary and are unable to perform *analogical innovation*, despite it being a key mechanism in the evolution of languages [29, 30].

2.2.1. Retrieval

Analogy solving on word embeddings has been around since early works on *Latent Semantic Analysis* [31] and word embeddings [20, 23], in which examples like $king - man + woman = queen$ have been used do demonstrate the ability to encode semantic features in the word representation. These examples can be formulated as analogical equations $man : woman :: king : X$, for which the solution is retrieved among a *vocabulary* of candidate words. In [23], the authors use morphological[1] analogies to demonstrate that some word embedding models encode a degree of morphological information. Two of the most used methods for solving analogies in embedding spaces by retrieval are 3CosAdd [20] and 3CosMul [32]. In 3CosAdd, the solution X is retrieved from the vocabulary by minimizing the cosine distance $cos(e_{word}, e_X)$, with $e_X = e_C - e_A + e_B$ and e_A, e_B, e_C, and e_X the embeddings of A, B, C, and X. 3CosMul follows a similar intuition but we refer the reader to [32] for a detailed description. However, the quality of the solution produced by the methods described above have been criticized by [19] for being far from human performance in some cases. Nonetheless, frameworks based on analogy datasets like those mentioned in [8] appear to bridge this gap in performance. By replacing the arbitrary formula by a learned estimator, Lim et al. significantly improved performance on solving semantic word analogies. This latter approach was adapted to morphological word analogies in [25] and outperforms the generative methods described in Subsubsection 2.2.2. Those two approaches rely on the postulates of analogical proportions, and achieve high analogy solving performance.

While the approach by Marquer et al. has state of the art performance on solving analogical equations in morphology, it suffers from the limitations of retrieval approaches: the solutions are retrieved from a *de facto* finite vocabulary and analogical innovation is impossible. By using a generative deep learning model, the present work aims to maintain state of the art performance while solving the limitation of retrieval approaches.

2.2.2. Generation

In [33], the author uses the postulates of [26] to address multiple characteristics of words, such as their length, the occurrence of letters and of patterns. Based on these features, Lepage proposes an algorithm to solve analogies between character strings. Following the results of [34] about closed form solutions, the *Alea* algorithm [6] proposes a Monte-Carlo estimation of the solutions of an analogical equation by sampling among multiple sub-transformations. Those sub-transformations are obtained by considering the words as bags of characters and generating permutations of characters that are present in B but not in A on one side, and characters of C on the other. Intuitively, if we consider $bag(A)$ the bag of characters in A, *Alea* considers $bag(D) = (bag(B) - bag(A)) + bag(C)$ and thus D is a permutation of the characters of $bag(D)$. Recently, a more empirical approach was proposed by [9], which does not rely on the axioms of analogical proportions. The generation model proposed by the authors considers some transformation f such that $B = f(A)$ and $f(C)$ is computable. The simplest transformation f is usually the one human use to solve analogies [9], and is found by minimizing the Kolmogorov complexity of f. This complexity is estimated by first expressing f using a language of operations (insertion, deletion, *etc.*), and computing the length of the resulting program. Unlike *Alea*, *Kolmo* is able to handle mechanisms like reduplication (repeating part of a word).

Recently, [22] proposed a generation framework to solving sentence analogies. They use an autoencoder model (named ConRNN) trained to reconstruct sentences, and perform simple

[1] In [23] the authors refer to morphological transformations as *syntactic* transformations, because they refer to the syntactic role of the word (*e.g.*, past participle) and not the arrangement of its morphemes (*e.g.*, the addition of the suffix "-ed").

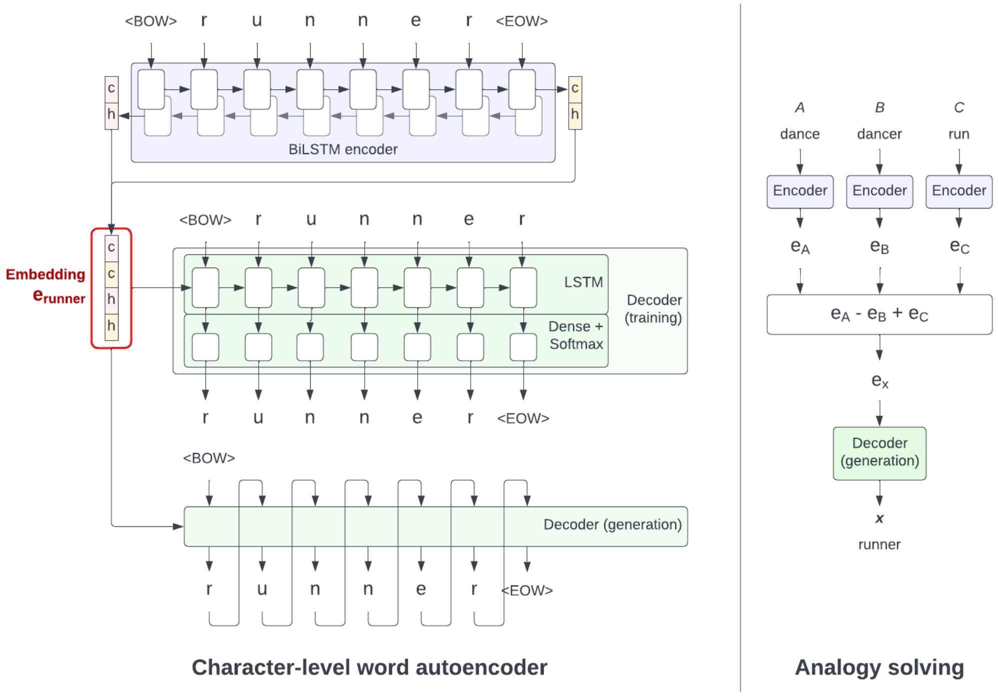

Figure 1: Character-based word auto encoder and vector arithmetic to solve analogies

arithmetic operations on the embedding space to solve analogies. Once the analogy between embeddings is solved, the decoder part of ConRNN is used to generate the solution from the predicted embedding. Their model is a sequence-to-sequence model composed of 2 elements. First, a sentence (as a sequence of words) is used as input to an encoder RNN, and the last hidden state of the RNN is used as the sentence embedding. The latter is then fed to a decoder RNN that tries to predict the words of the input sentence. The use of a generative model achieves significantly better results than previous retrieval approaches on the same embedding space. The current work aims to extend the one of [25] by replacing the retrieval by the generation of the solution of morphological analogical equations. To do so, it is necessary to generate words at the character level from fixed-size embeddings, however in the literature there is to our best knowledge no approach proposed to tackle this specific issue. Inspired by the success of [22], we propose a character-level autoencoder for words and display its performance in solving morphological analogies.

3. Our Approach

In this section we present the approach we use, illustrated in Figure 1. The architecture for our model is a character-level sequence-to-sequence autoencoder model, based on the model described in [35]. In order to properly decode the final vector solution, the model is trained to encode words and then decode the resulting vector back into the same word. Each character in a word w is encoded into a one-hot vector and is then fed into the encoder, which uses a Bidirectional Long Short Term Memory (BiLSTM) layer. This layer outputs four vectors: the last hidden state h_f and cell state c_f in the forward direction, and similarly h_b and c_b for the backward direction. The concatenation of these vectors $e_w = \text{concat}(h_f, h_b, c_f, c_b)$ is the embedding of the word. The decoder is a regular LSTM layer, followed by a dense layer with softmax activation. The input for the first step of the decoder is the above-mentioned embedding, split into two states $h = \text{concat}(h_f, h_b)$ and $c = \text{concat}(c_f, c_b)$. During training, we use *teacher forcing*: (*i*) the characters of the word w to predict are used as input, with an added *beginning-of-word* (BOW) character at the begining; (*ii*) the prediction targets are the characters of w, but ahead by one time-step and with an *end-of-word* (EOW) character at the end.

To compute the solution of an analogy $A : B :: C : X$, the embeddings e_A, e_B, and e_C are computed by the encoder and used to compute $e_X = e_B - e_A + e_C$. Then, e_X is decoded into a word X by the decoder. Beginning with the BOW character, at each time-step the sampled character with the highest probability of occurrence is added to the word until either the EOW character is predicted or the length of the word is the same as the longest word in the dataset.

4. Experiments

In this section we present our experimental setup. First, the dataset we use is described in Subsection 4.1. We then report the performance of our model in the autoencoder setting in Subsection 4.2. The analogy solving performance of our approach is compared with baselines in Subsection 4.3. Finally, we discuss the overall performance of the model in Subsection 4.4.

4.1. Datasets

For our experiments, we used the analogies from 8 languages available in the Siganalogies dataset [36]: Arabic, English, French, German, Hungarian, Portuguese, Russian, and Spanish extracted from the high resource languages of Sigmorphon2019 [37]. These languages were chosen such that, in later stages of the work, the authors have enough linguistic knowledge to interpret the model outputs. In order to obtain train and test sets, non-overlaping random subsets of analogies from the entire Sigmorphon dataset for a given language are taken, to ensure that no analogy is seen in both the train and test sets.

The Siganalogies dataset also provides a method for data augmentation via permutating the four words in a given analogy. These permutations are obtained using the *symmetry* and *central permutation* postulates of analogy. From a *base form* $A : B :: C : D$, we generate 7 permutations:

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

Table 1
Autoencoder accuracy at the word level for 8 languages, trained for 100 epochs on 40,000 random words.

Language	Accuracy (%)
Arabic	99.99
English	99.98
French	99.99
German	99.98
Hungarian	99.97
Portuguese	99.99
Russian	99.96
Spanish	99.98

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

In Siganalogies, the base forms $A : B :: C : D$ are such that B is an inflected form of A and D is inflected from C. In addition to that, base forms $A : A :: B : B$ derived from the identity postulate ($A : A :: B : B$ is true for all A and B) are present. An example of analogy in English is *dog* : *dogs* :: *cat* : *cats*, another in French is *revérifier* : *revérifiasse* :: *tormenter* : *tormentasse*, and in German there is *Donor* : *Donor* :: *Herstellungsverfahren* : *Herstellungsverfahren* (identity, but also accusative singular declension of the noun).

4.2. Autoencoder performance

As shown in Table 1, our autoencoder achieves very high accuracy in decoding vectors back into words, meaning that any wrong solutions are a result of the operations performed on the analogy rather than the decoding process. In our experiments, the model encodes the words as 128-dimensional vectors and there is a 0.1 dropout on the decoder LSTM layer. The loss function used is categorical cross-entropy, since a probability for the likelihood of each character appearing is required at each time-step. An 80/20 train/validation split is used, and the validation loss was the metric used for the early stopping. If the early stopping is not triggered, the model is trained for 100 epochs.

4.3. Analogy solving performance

Two metrics are used to determine the performance of our model. The first one is a variation of Levenshtein distance, which calculates the minimum number of edits required to change one sequence into another using insertions, deletions, and substitutions. In order to display how close the decoded analogy solutions are to the expected analogy solutions, the Levenshtein distance L was normalized using the length of the manipulated words into a percentage L_p like so:

$$L_p(expected, decoded) = 1 - \frac{L(expected, decoded)}{\max\left(len_{expected}, len_{decoded}\right)}$$

Table 2
Results for 8 languages for 10,000 base analogies and all of their permutations (80,000 analogies in total). We report L_p in % and the accuracy (Acc.) in %. Our autoencoder was trained for 100 epochs on 40,000 random words per language. Baselines Alea [6] and Kolmo [9] were tested in the same setting. The accuracy of the retrieval model ANNr [25] is reported as mean ± standard deviation for 10 random initialization, but note that these results are not completely comparable with our approach as they were obtained in a closed setting.

Language	Score	Ours	Alea	Kolmo	ANNr
Arabic	L_p	**54.51**	23.72	45.31	–
	Acc.	**12.50**	2.56	3.81	71.80 ± 2.51
English	L_p	**91.58**	88.34	86.75	–
	Acc.	**59.80**	59.65	46.93	94.40 ± 0.67
French	L_p	86.43	80.07	**89.32**	–
	Acc.	51.30	**57.64**	54.49	91.84 ± 0.83
German	L_p	**89.39**	82.76	87.47	–
	Acc.	**52.80**	50.84	48.97	76.95 ± 1.15
Hungarian	L_p	**80.32**	60.72	75.47	–
	Acc.	25.50	**27.80**	23.48	80.42 ± 1.30
Portuguese	L_p	**94.38**	87.97	93.47	–
	Acc.	74.00	**80.06**	71.28	89.30 ± 2.38
Russian	L_p	82.29	63.52	**82.78**	–
	Acc.	33.80	**37.15**	33.44	72.65 ± 1.96
Spanish	L_p	**89.39**	79.49	88.56	–
	Acc.	60.09	**65.02**	58.59	93.01 ± 2.38

The resulting percentage measures the rate of correctly decoded characters per word - when it is 1 (or 100%), then the decoded solution matches the expected solution perfectly. The second metric, accuracy, was calculated by dividing the number of correctly decoded analogies by the total number of analogies for each language. We report the results of decoding a test set of 10,000 base analogies and their permutations in Table 2. We compare our approach with Alea [6] and Kolmo [9] described in Subsubsection 2.2.2. We also report the retrieval accuracy of ANNr [25], however as the model is a retrieval approach it is not directly comparable to our model and other baselines. Instead, it indicates the performance one can reach when bypassing the issue of generation. Our model reaches comparable performance to the generation baselines in terms of L_p for all languages, and comparable performance in terms of accuracy for half of the languages.

4.4. Discussion

The performance on Arabic of all generation models is very low, while the retrieval model does not appear to suffer from the same effect. Further analysis of the data reveals that the character encoding used for Arabic decomposes each character into multiple encoded characters, resulting in longer and more complex sequences of characters than expected. We suppose this makes generation harder and is the cause of this low performance.

There is a significant difference in the performance of the model depending on which permutations are used. Due to the high accuracy of the decoder and the nature of the parallelogram rule, the model performs very well on analogies where the solution D is the same as another element in the analogical equation. Permutations of this form include strong reflexivity, strong inner reflexivity, and identity.

As Table 2 shows, the raw accuracy is often lower than the baselines Alea and Kolmo, but the Levenshtein percentage is often on par or higher. This suggests that more individual characters are correctly decoded with our model on average when compared to the baselines, but that it does not decode entire words with 100% accuracy as often. This is to be expected as the model is not trained to solve analogies, but rather is trained to properly decode words after vector arithmetic is performed on the encoded vectors.

When applied to the encoded vectors, the parallelogram rule is highly accurate with regular morphology and with certain permutations, but it often struggles when the morphology is more irregular. Since the model decodes individual words with high accuracy, the problem lies with the operations performed on the three vectors in an analogical equation after encoding. Given the model's current performance without explicitly encoding any morphological features or features of analogical equations when training, we expect that better performance can be obtained if these features are included in future iterations of the trained autoencoder.

5. Conclusion and Perspectives

In this paper we proposed an autoencoder framework to solving morphological analogies by generating solutions. This partially addresses the limitations of previous works relying on case based approaches that prevent creative inference and innovation. Our adaptation to the morphology setting was illustrated in several languages with promising results, and that reveal new potential directions for future work.

However, this is a preliminary proposal that will profit from further training and the combination with state of the art retrieval approaches such as ANNr from [25]. Moreover, we will also explore its tranferability potential and its generalization across multiple modalities and data contexts.

Acknowledgments

This research work 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. A. Fahandar, E. Hüllermeier, Learning to rank based on analogical reasoning, in: AAAI, 2018, pp. 2951–2958.

[2] M. A. Fahandar, E. Hüllermeier, Analogical embedding for analogy-based learning to rank, in: IDA, volume 12695 of *LNCS*, Springer, 2021, pp. 76–88.

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

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

[5] M. Couceiro, N. Hug, H. Prade, G. Richard, Analogy-preserving functions: A way to extend boolean samples, in: 26th IJCAI, 2017, pp. 1575–1581.

[6] 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.

[7] R. Fam, Y. Lepage, Morphological predictability of unseen words using computational analogy., in: 24th ICCBR workshops, 2016, pp. 51–60.

[8] S. Lim, H. Prade, G. Richard, Solving word analogies: A machine learning perspective, in: 15th ECSQARU, volume 11726, 2019, pp. 238–250.

[9] 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.

[10] F. Sadeghi, C. L. Zitnick, A. Farhadi, Visalogy: Answering visual analogy questions, in: NeurIPS, 2015, pp. 1882–1890.

[11] J. Peyre, I. Laptev, C. Schmid, J. Sivic, Detecting unseen visual relations using analogies, in: IEEE ICCV, 2019, pp. 1981–1990.

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

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

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

[15] C. Antic, Analogical proportions (2022).

[16] N. Barbot, L. Miclet, H. Prade, Analogy between concepts, Artificial Intelligence 275 (2019) 487–539.

[17] 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.

[18] D. E. Rumelhart, A. A. Abrahamson, A model for analogical reasoning, Cognitive Psychology 5 (1973) 1–28.

[19] A. Drozd, A. Gladkova, S. Matsuoka, Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen, in: 26th COLING, 2016, pp. 3519–3530.

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

[21] D. Chen, J. C. Peterson, T. Griffiths, Evaluating vector-space models of analogy, in: 39th CogSci, Cognitive Science Society, 2017, pp. 1746–1751.

[22] L. Wang, Y. Lepage, Vector-to-sequence models for sentence analogies, in: ICACSIS, 2020, pp. 441–446.

[23] T. Mikolov, W.-T. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: NAACL, 2013, pp. 746–751.

[24] 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.

[25] E. Marquer, S. Alsaidi, A. Decker, P.-A. Murena, M. Couceiro, A Deep Learning Approach to Solving Morphological Analogies, 2022. URL: https://hal.inria.fr/hal-03660625.

[26] Y. Lepage, S. Ando, Saussurian analogy: a theoretical account and its application, in: 16th COLING, 1996.

[27] 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.

[28] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: EMNLP, 2014, pp. 1532–1543.

[29] D. L. Fertig, Analogy and morphological change, Edinburgh University Press, 2013.

[30] E. Mattiello, Analogy in Word-formation, De Gruyter Mouton, 2017.

[31] 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.

[32] O. Levy, Y. Goldberg, Dependency-based word embeddings, in: 52nd ACL (Volume 2: Short Papers), ACL, 2014, pp. 302–308.

[33] Y. Lepage, Character-position arithmetic for analogy questions between word forms, in: 25th ICCBR workshops, volume 2028, 2017, pp. 23–32.

[34] F. Yvon, Finite-state transducers solving analogies on words, Rapport GET/ENST&LTCI (2003).

[35] F. Chollet, Character-level recurrent sequence-to-sequence model, 2017. URL: https://keras.io/examples/nlp/lstm_seq2seq/.

[36] E. Marquer, M. Couceiro, S. Alsaidi, A. Decker, Siganalogies - morphological analogies from Sigmorphon 2016 and 2019, 2022.

[37] 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.

Exploring Analogical Inference in Healthcare

Safa Alsaidi[1,2,*,†], Miguel Couceiro[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 $A : B :: C : D$ that are used to map similar relationships between two pairs of objects, A, B, and C, D. Analogies have long been a subject of research in the Natural Language Processing (NLP) community, where they have been applied to a variety of reasoning and classification tasks. Lately, machine and representation learning have shown to be useful for analogical reasoning. In this paper, we discuss the possibility of adapting the analogical framework to healthcare applications, in particular to medical decision support. We particularly hypothesize that as language representations help in analogical reasoning in NLP, patient representation learned from Electronic Health Records (EHRs) may help in healthcare. We define three different analogy based settings adapted to EHR data that we see as first steps to the development of analogical applications to this domain. We provide statistics on the first sets of analogies that we built from a publicly available dataset of EHRs, and report preliminary, but promising results to detect patient-stay analogies following our very first experimental setting.

Keywords

analogy classification, electronic health records, patient representation learning

1. Introduction and motivation

An analogical proportion, or an analogy, is a relation between four objects A, B, C, and D that is expressed as "A is to B as C is to D" and formally denoted as $A : B :: C : D$. There are two main tasks associated with analogical proportions: *analogy detection* and *analogy solving*. *Analogy detection* corresponds to the task of deciding whether a quadruple $\langle A, B, C, D \rangle$ is a valid analogy. *Analogy solving* corresponds to finding a fourth element x so that $A : B :: C : x$ is a valid analogy. This can be done either by retrieving x from a pool of candidates or by

IARML@IJCAI-ECAI'2022: Workshop on the Interactions between Analogical Reasoning and Machine Learning, at IJCAI-ECAI'2022, July, 2022, Vienna, Austria

[*] Corresponding author.

safa.alsaidi@inria.fr (S. Alsaidi); miguel.couceiro@loria.fr (M. Couceiro); 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); 0000-0002-4782-6737 (S. Quennelle); 0000-0001-6855-4366 (A. Burgun); 0000-0002-3326-2811 (N. Garcelon); 0000-0002-1466-062X (A. Coulet)

CEUR Workshop Proceedings (CEUR-WS.org)

generating x. Analogies have been extensively studied and applied to various Natural Language Processing (NLP) tasks [1, 2, 3, 4]. Object representations called *embeddings* are low-dimensional representations of high-dimensional vectors, which have been used to improve deep learning methodologies. Some of these embeddings learn precise representations and are able to detect differences between objects. As a result they can discriminate between valid and invalid analogical proportions and solve analogical equations.

In this paper, we explore the possibility to leverage the analogy framework to solve tasks relevant to the healthcare domain. We particularly consider using Electronic Health Records (EHRs) to learn patient representations, *i.e.*, patient embeddings. We initiate the construction of sets of patient-based analogies using relationships existing between patient hospital stays from a publicly available set of EHRs. These health records consist of clinical and administrative data collected during patient hospital stays. Generally they are composed of structured (*e.g.*, diagnostic codes, lab tests) and unstructured data (*e.g.*, clinical notes, nursing reports, discharge summaries), either static (*e.g.*, patient demographics) or temporal (*e.g.*, vital signs).

EHRs have been secondary used to conduct epidemiological and observational studies. They have also been used as real word data to train predictive models [5]. In particular, deep learning methods have become increasingly popular in medical informatics for general tasks such as predicting mortality, in-hospital readmission, diagnoses, etc. A key element for such tasks is to effectively convert patient data from the raw EHR format to embeddings that can be further processed [6]. Representation learning thus consists of learning low-dimension feature representations from raw data. As EHR data are heterogeneous and complex, studies have shown that deep learning models are suited to encode complex EHR data to learn patient representations and that various architectures are suited to different biomedical tasks [7, 8, 9, 10, 11, 12]. For instance, Madhumita et al. [13] used a stacked denoised autoencoder and a paragraph vector model to learn generalized patient representations directly from clinical notes. Si and Roberts [14] utilized a three-level hierarchical attention-based recurrent neural network (HAN) with greedy segmentation to learn patient representation from clinical notes. Zhang et al. [15] proposed 2 multi-modal neural network architectures to enhance patient representation learning by combining sequential unstructured notes with structured data.

Analogies have only been sporadically applied to healthcare. Nonetheless, analogical reasoning has been applied in clinical practice by physicians for diagnosis and prognosis, as a way of linking visible signs and symptoms to possible causes. Indeed, medical reasoning relies on observations of previous patients with similar signs and symptoms, who happened to have a certain disease. Several studies have investigated analogies in healthcare by applying various machine learning methods. For instance, Rather et al. [16] used analogical proportions to identify hidden or unknown biomedical knowledge from free text resources. In their work, they defined analogies of the form "*acetaminophen* is a type of *drug* as *diabetes'* is a type of *disease*." Dynomant et al. [17] used analogical proportions to compare embedding methods trained on a corpus of French health-related documents. Each analogical proportion aimed to verify if $(\vec{Term}1 - \vec{Term}2) + \vec{Term}3 \approx \vec{Term}4$, allowing to check if the similarity between the first two terms is similar to the one between Term 3 and Term 4.

In this paper, we describe an ongoing work on analogical inference in healthcare. We introduce three analogy based settings, where each setting aims to investigate specific biomedical tasks, namely identity, predictive, and generative tasks. In comparison with previous studies

[14, 7, 8, 9, 10, 11, 12], we aim to build analogies based on patient-stay representations. One of the main contributions of our work is a framework to build sets of proportions for analogical inference in healthcare.

This paper is organized as follows. Section 2 provides a description of the MIMIC-III dataset. Section 3 defines our analogical settings and associated biomedical tasks, and justifies our task choices. Section 4 presents preliminary statistics of the analogical proportions built from MIMIC-III. Section 5 initiates a discussion addressing some analogical postulates that could be useful when generating our analogies. Section 6 illustrates the feasibility of our approach by providing preliminary results using one of our experimental settings. Section 7 discusses perspectives for future research.

2. Data description

We propose to use EHRs as a source of patient medical history data and aim to consider both its structured and unstructured data to define our analogies. In particular we experiment with a publicly available dataset of EHRs called MIMIC-III (Medical Information Mart for Intensive Care-III) [18]. MIMIC-III is a critical care database, developed by the Massachusetts Institute of Technology (MIT)'s Laboratory for Computational Physiology and distributed by PhysioNet [19]. It contains integrated, de-identifed health-related data in accordance with Health Insurance Portability and Accountability Act (HIPAA). It contains data associated with all patients admitted to the ICU (Intensive Care Unit) of Beth Israel Deaconess Medical Center between 2001 and 2012. It contains various data, such as patient demographics, vital signs, lab test results, medications, hospital length of stay, survival, clinical notes, imaging reports and more, structured into 26 tables. Each patient-stay is associated with diagnosis codes, motivating the stay and procedures performed during the stay. It encompasses data of more than 40,000 ICU patients and more than 60,000 ICU stays. Table 1 shows statistics for the subgroups of adult patients (aged 18 and above) with at least two stays, which is the subset that we consider in the rest of the article.

The database contains a combination of structured and unstructured data and is accessible to researchers under a data use agreement, where users are required to follow a HIPAA training course demanded by the National Institutes of Health (NIH).

	Statistics
Patients (total)	$8,526$
Gender, male (total)	$4,818$
Age (median, in years)	66.24
ICU stays (total)	$23,345$
Hospital stays (total)	$19,709$
ICU length of stay (median, in days)	2.33
Hospital length of stay (median, in days)	9.74
Clinical notes per stay (median)	18.0

Table 1
General statistics of the MIMIC-III EHR dataset, restricted to patients aged of 18 and above, with at least 2 stays.

3. Experimental settings and biomedical tasks

As we defined previously, an analogy is a 4-ary relation and is usually written as $A : B :: C : D$. In this paper, we define three analogy based settings and associated tasks that we are interested in investigating with EHR data. We name our three settings as follows: *(i)* Identity; *(ii)* Identity + Sequent; *(iii)* Identity + Directly Sequent. For these settings, we do not want to learn "full" patient representation, but *patient-stay representations* (*i.e.*, learn a numeric vector representation of EHR data that belong to a single hospital stay) which we hope to be simpler.

Identity In the first 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 could be the same patient, and that t_1 and t_2, or t_3 and t_4, could represent the same time stamp. Furthermore, t_1 and t_3 or t_2 and t_4 could be the same when $i_1 = i_2$ (but not when $i_1 \neq i_2$). In this setting we aim at investigating identity tasks, *i.e.*, associating an unaffected sample of data to the patient it belongs. Note that this setting fits several data cleaning and data privacy related applications

Identity + Sequent For this setting, we add a temporal constraint to analogies, as we force

$$s_{t_1}^{i_1} \ll s_{t_2}^{i_1} \text{ and } s_{t_3}^{i_2} \ll s_{t_4}^{i_2}$$

where $\ll$ denotes temporal sequentiality between stays of a same patient, *i.e.*, $s_{t_2}^{i_1}$ takes place after $s_{t_1}^{i_1}$ and $s_{t_4}^{i_2}$ takes place after $s_{t_3}^{i_2}$ but not necessarily directly right after. We consider cases where $i_1 = i_2$. In this setting, we also define a relation named **diagnosis**, which forces $s_{t_1}^{i_1}$ and $s_{t_3}^{i_2}$ to have the same diagnosis. This relation provides more meaning to our analogies and gives us more medical insight into the relationship between our patients. For example, based on the different stays associated with a single patient we hope to see how a certain disease develops (similarly or differently) between two distinct patients.

Identity + Directly Sequent In this third setting, we make the temporal constraint more strict as we force the two stays of the same patient to be directly sequent (no other stay can exist in between). We note this constraint

$$s_{t_1}^{i_1} \prec s_{t_2}^{i_1} \text{ and } s_{t_3}^{i_2} \prec s_{t_4}^{i_2}$$

The **diagnosis** relation is kept between $s_{t_1}^{i_1}$ and $s_{t_3}^{i_2}$, and cases where $i_1 = i_2$ are also considered. With these three settings, we aim at investigating the applicability of two tasks: *analogy detection* and *analogy inference*. For instance, given an analogy of the form $A : B :: C : x$, we can either propose potential values for an unknown stay x (*i.e.*, predictive task) or generate stays which would enrich our dataset with synthetic stays (*i.e.*, generative task).

For the last two settings, we define additional settings by considering three levels of relaxation of the diagnosis constraint. It is satisfied either if both stays are associated with the very same primary diagnostic code (level 4) or in more relax settings, *i.e.*, if both codes belong to the same level-3 or level-2 branch of the hierarchy of the ICD-9-CM (International Classification of Diseases, Ninth Revision, Clinical Modification) [20]. Statistical details on the influence of the constraints is discussed in the next section.

4. Preliminary statistics on MIMIC-III

We computed some preliminary statistics on MIMIC-III dataset to check how many analogical proportions can be formed for each of the three analogical settings and based on the three level diagnosis constraint as shown in Table 3. To form the analogies, we built tuples of each of the two stays that belong to a single patient i. We kept only adult patients (aged 18 and above) that have at least two hospital stays. For our first setting, we define a *valid analogy* as a quadruple of four stays $(s^{i_1}_{t_1}, s^{i_1}_{t_2}, s^{i_2}_{t_3}, s^{i_2}_{t_4})$, where each pair of two stays belong to a single patient i_j. Since we do not restrict the order of the stays for each of the pairs, our analogies were made of all the permutations of all the stays belonging to a patient.

For our second and third settings, we define a *valid analogy* to be a quadruple made of four stays $(s^{i_1}_{t_1}, s^{i_1}_{t_2}, s^{i_2}_{t_3}, s^{i_2}_{t_4})$, where each pair of two stays belong to a single patient i_j and $s^{i_1}_{t_1}$ and $s^{i_2}_{t_3}$ have the same diagnostic code. As an order constraint is introduced for these two settings, we had to make sure that $s^{i_1}_{t_1}$ takes place before $s^{i_1}_{t_2}$ and $s^{i_2}_{t_3}$ takes place before $s^{i_2}_{t_4}$. For the second setting, the stays do not necessarily happen directly right after, where other stays can exist in between. As for the third setting, one stay immediately follows the other, *i.e.*, there is no other stay in between them.

For the diagnosis constraint, we referred to the ICD-9-CM, which is the standard nomenclature for assigning diagnosis codes to each hospital stay. Indeed, each stay has a unique primary diagnosis code and a set of secondary codes. Diagnostic codes are organized hierarchically as follows: *(1) chapter*, *(2) block*, *(3) 3-digit category*, and *(4) full code.* As an example, the diagnosis code 767.4 and its hierarchy are presented Table 2.

Level	Level name	Example of code range
1	Chapter	760–779
2	Block	764–779
3	3-digit category code	767
4	Full code	767.4

Table 2
Hierarchy of the ICD9-CM (International Classification of Disease, Ninth Revision, Clinical Modifications), with examples for the code 767.4 ("Injury to spine and spinal cord due to birth trauma").

The MIMIC-III dataset associates each stay with an ICD-9 code (*i.e. full code*). For the second and third analogical settings, we preprocessed our diagnosis codes in the following manner. We kept only the primary diagnostic code associated with each patient-stay. We filtered ICD-9 codes that appeared only once. For the second level and third level diagnosis settings, we performed the same preprocessing except that we had to first convert the ICD-9 codes into their corresponding *category* and *block* formats.

Table 3 provides the number of *valid* analogies that can be formed with the defined settings. As shown, the number of analogies is the highest for the *Identity* setting, which could be explained as a result of the absence of constraint on both the order of stays and diagnosis. The more strict the order constraint is, the less the amount of *valid* analogies that could be formed. The *diagnosis* constraint also influences the number of analogies that could be generated. The number of analogies is the lowest for *full code* (level 4) settings, where less patients share the very same primary diagnosis code. In comparison, more patients can share a single diagnosis code in the *category* (level 3) and *block* (level 2) settings, where we observe the highest number of analogies in the *block* setting for both the second and third analogical settings.

Setting	ICD Level	Analogies
Identity	N/A	$1,100,954,350$
Identity + Sequent	4	$648,169$
	3	$1,876,445$
	2	$3,243,699$
Identity+Directly Sequent	4	$545,892$
	3	$1,326,518$
	2	$2,780,507$

Table 3
Number of valid analogies that can be generated from MIMIC-III, depending on the different settings and on the level of flexibility allowed on patient ICD diagnosis.

5. Properties of analogies for data augmentation

As we are interested in exploring different deep learning models, we would need large amounts of data to train them. To enlarge the training datasets, one may use analogy properties to generate more analogies in a process called *data augmentation.* Training our model on different equivalent forms of the same analogy could help reduce overfitting. Previous works [21, 2, 22] have defined postulates that proportional analogy should obey; some of which include the following:

- reflexivity: $A : B :: A : B$
- inner reflexivity: $A : A :: C : C$
- determinism: $A : A :: A : D \to D = A$
- symmetry: $A : B :: C : D \to C : D :: A : B$
- inner symmetry: $A : B :: C : D \to B : A :: D : C$
- central permutation: $A : B :: C : D \to A : C :: B : D$.

However, not all these postulates hold for all our settings. Based on the current definitions of our analogical settings, we can apply *reflexivity* for all the three settings. *Inner reflexivity* can only be applied for the *Identity* setting. Adding this postulate for the second and third settings would require to loose our order constraint, which is inconsistent with the temporal aspect of predictive modeling. *Determinism* holds for all the settings. We include this postulate even if it produces trivial analogies. *Central permutation* can be applied on our analogies for the first setting only and 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. Concerning the second and third analogy settings, *central permutation* cannot be applied as it violates the order constraint in most cases. Note that *central permutation* can be applied for these two settings for cases when $i_1 = i_2, t_2 \leq t_1, t_3 \leq t_4$, and the same diagnosis is associated to $s^{i_1}_{t_1}$ and $s^{i_2}_{t_3}$. *Inner symmetry* can be applied for the *Identity* setting, but it violates the order constraints for the other two settings. For all the three analogical settings, by applying *symmetry* to one valid analogy, we can increase the number of *valid* analogies as it does not violate any of the three constraints. In addition to valid forms, we can also consider *invalid* forms (*i.e.*, that contradict some of the setting constraints or that cannot be inferred from the base cases using the allowed postulates) for classification purposes.

6. Preliminary experiments: error analyses in the Identity setting

We set up a preliminary experiment on the analogy detection task, addressing our *Identity* setting. Inspired by [3, 23], we consider a CNN classifier adapted to patient-stay, to determine whether a given (A, B, C, D) constitutes a valid analogy. For the embedding model, we consider the Fusion CNN model developed by [15], which combines both structured and unstructured data to obtain patient-stay representations. For this very first experiment, we only consider structured data limited to demographics and admission-related information. For unstructured data, we group clinical notes associated with a hospital stay. We learn clinical note embeddings and concatenate them with static information following [15] to obtain our final patient-stay representations.

We considered hospital stays of 200 patients extracted randomly from MIMIC-III. We define a *valid analogy* as a quadruple of four stays $(s^{i_1}_{t_1}, s^{i_1}_{t_2}, s^{i_2}_{t_3}, s^{i_2}_{t_4})$, where each pair of two stays belong to a single patient i_j. We do not define any order constraint for our stays; therefore, $s^{i_1}_{t_1}$ can happen before $s^{i_1}_{t_2}$. Quadruples where $i_1 = i_2$ are also included in the dataset. To generate other valid analogies, we make use of all postulates in Section 5, except for *central permutation* that is only applied in the case when $i_1 = i_2$. As *reflexivity* forces $i_1 = i_2$, it cannot be applied in the cases where $i_1 \neq i_2$. Accordingly, given a valid analogy $A : B :: C : D$, we generate 8 additional valid analogies, 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 2 invalid, namely, $D : A :: B : C$ and $A : C :: B : D$. When $i_1 = i_2$, we generate one more valid analogy of the form $A : B :: A : B$ and we consider invalid analogies as valid.

For training and evaluation, we split our dataset into 70% training set and 30% testing set, representing 939,638 analogies for training and 402,703 for testing. We randomly draw 50,000 analogies of each set (*i.e.*, training and testing) when loading the data. Following the data augmentation procedure introduced before, given a valid analogy, we generate 9 valid analogies (*i.e.*, positive examples) and 2 invalid analogies (*i.e.*, negative examples) for cases when $i_1 \neq i_2$. In contrast, we generate 12 valid analogies and no invalid analogies for cases when $i_1 = i_2$. Based on this setting, we tend to generate more valid analogies than invalid ones. We trained our model on 10 epochs, with 3 random intializations to observe how the model behaves and how much it is able to learn. We only computed the accuracy and obtained 96.85 ± 1.75 for valid analogies and 70.31 ± 1.94 for invalid analogies. Our model performs the best for positive examples which can be explained as a result of the imbalance between valid and invalid examples in the training data. Nonetheless, these preliminary results seem to show that the model learns, to some extent, patient-stay identity relationships.

To gain a deeper insight on how our classification model works, we present four examples: one true positive, one false negative, one true negative, and one false positive. Patient ids in the examples below have been changed and dates have been shifted. We provide elements of interpretation to explain why our model correctly classifies some analogies and why in other cases it does not.

Analysis of a true positive example. We consider the stay $s_{t_1}^{i_1}$ of patient 1249, who is a female, with 83yo, suffers from Measles keratitis, admitted twice before, and with 12 Radiology reports documenting this stay. The second stay $s_{t_2}^{i_1}$ of the same patient 1249, but with 81yo, suffers from Pancreat cyst/pseudocyst, only admitted once before, and with 5 Radiology reports and 7 Nursing/Other reports. The stay $s_{t_3}^{i_2}$ belongs to patient 4695, who is a female, 21yo, with Acute venous embolism and thrombosis of superficial veins of upper extremity, admitted 5 times before, and with 5 Radiology reports and 2 Nursing/Other reports. The stay $s_{t_4}^{i_2}$ of the same patient 4695, but with 22yo, suffers from Hypertensive chronic kidney disease, admitted 8 times before, and with 5 Physician reports and 7 Nursing reports documenting this stay.

This example has been correctly classified as *valid* for all the 9 valid forms. As we do not introduce any order constraint for this setting, we can notice that for some forms like $A : B :: C : D$ and $A : B :: D : C$, $s_{t_2}^{i_1}$ would take place before $s_{t_1}^{i_1}$ in time. The model correctly classifies these analogy forms as valid.

Analysis of a false negative example. We consider the stays in this example to belong to the same patient 1109, who is a female. The stay $s_{t_1}^{i_1}$ of patient 1109, with 25yo, suffers from Malignant essential hypertension, admitted 7 times before, and with 1 Radiology report and 2 Nursing/Other reports documenting this stay. The second stay $s_{t_2}^{i_1}$ of patient 1109, but with 26yo, with Hypertensive chronic kidney disease, admitted 4 times before, and with 8 Physician reports and 4 Nursing reports. The third stay $s_{t_3}^{i_2}$ of patient 1109, but with 26yo, admitted once again for Hypertensive chronic kidney disease, admitted 3 times before, and with 3 Physician reports and 8 Nursing reports. The fourth stay $s_{t_4}^{i_2}$ of patient 1109, with 27yo, suffers from Vascular complications of medical care, admitted 5 times before, and with 3 Physician reports and 9 Nursing reports.

As we mentioned above, for cases where $i_1 = i_2$, applying central permutation would also give us valid analogies. In this example, our model incorrectly classified the form of $D : A :: B : C$ as invalid. As there were less analogies made of four stays that belong to the same patient included in our dataset, we noticed that our model is more likely to incorrectly classify these analogies, particularly for invalid forms.

Analysis of a true negative example. We consider the stay $s_{t_1}^{i_1}$ of patient 553, who is a male, with 23yo, suffers from Hypertensive chronic kidney disease, admitted 4 times before, and with 4 Physician reports and 8 Nursing reports documenting this stay. The stay $s_{t_2}^{i_1}$ belongs to the same patient 553, but with 24yo, admitted once again for Hypertensive chronic kidney disease, admitted 6 times before, and with 6 Physician reports and 6 Nursing reports. The third stay $s_{t_3}^{i_2}$ belongs to patient 2387, who is a male, with 52yo, with Unspecified disease of pericardium, admitted 7 times before, and with 2 Radiology reports and 2 Nursing/Other reports. The stay $s_{t_4}^{i_2}$ belongs to the same patient 2387, but with 53yo, with Unspecified pleural effusion, admitted 9 times before, and with 8 Radiology reports and 4 Nursing/Other reports.

This analogy has been correctly classified as *invalid* for both invalid forms, $D : A :: B : C$ and $A : C :: B : D$.

Analysis of a false positive example. We consider the stay $s_{t_1}^{i_1}$ of patient 2771, who is a male, with 71yo, suffers from Subendocardial infarction, admitted 16 times before, and with 9 Nursing/Other reports. The second stay $s_{t_2}^{i_1}$ belongs to the same patient 2771, but with 70yo, suffers from Other pulmonary embolism and infarction, admitted 5 before, and with 1 Radiology report and 3 Nursing/Other reports. The stay $s_{t_3}^{i_2}$ of patient 2222, who is a female, with 69yo, with Diverticulosis of colon with hemorrhage, admitted 9 times before, and with 2 Physician reports and 10 Nursing reports. The stay $s_{t_4}^{i_2}$ belongs to the same patient 2222, but with 67yo, with Arterial embolism and thrombosis of lower extremity, admitted 5 times before, and with 3 Nursing/Other reports.

This analogy has been incorrectly classified as *valid* for the invalid form, $A : C :: B : D$. We noticed that when the category of the clinical notes is similar between two hospital stays and when our hospital stays do not include a lot of clinical notes, our model seems to struggle to distinguish between the two hospital stays. $s_{t_2}^{i_1}$ and $s_{t_4}^{i_2}$ have the same number of Nursing/Other reports. Thus these reports may not contain enough information to help our model differentiate between these two stays. As a result, the model incorrectly matches these stays to the same patient.

7. Conclusion

In this paper we discussed an exploratory approach to investigate analogical inference in healthcare. We started by briefly surveying some related work that address different applications of analogical reasoning in different domains. We defined three analogical settings for different healthcare tasks, and discussed the motivation behind our settings. We also presented preliminary statistics of the sets of analogical proportions that we built from MIMIC-III. The main contribution of our work is the formalization of settings that are meaningful in healthcare, and that guide the process of building sets of analogies in healthcare. We discuss the pertinence of certain widely used postulates in this healthcare context. Lastly, we also illustrated the *Identity* setting on which we addressed a preliminary experiment on the analogy detection task. These first results pave the way to conducting further experiments on the other two settings, and to an in depth analysis of the potential of coupling representation learning and analogical reasoning in healthcare.

Acknowledgments

We thank IARML reviewers for their constructive and positive feedback. 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 named author is partially supported by TAILOR, a EU Horizon 2020 project (GA No 952215), and the Inria Project Lab "Hybrid Approaches for Interpretable AI" (HyAIAI).

References

[1] 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, pp. 1848–1854.

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

[3] 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.

[4] 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.

[5] P. B. Jensen, L. J. Jensen, S. Brunak, Mining electronic health records: towards better research applications and clinical care, Nature Reviews Genetics 13 (2012) 395–405.

[6] Y. Si, J. Du, Z. Li, X. Jiang, T. A. Miller, F. Wang, W. J. Zheng, K. Roberts, Deep representation learning of patient data from electronic health records (ehr): A systematic review, Journal of biomedical informatics (2020) 103671.

[7] Y. Li, S. Rao, J. R. A. Solares, A. Hassaine, D. Canoy, Y. Zhu, K. Rahimi, G. Salimi-Khorshidi, Behrt: Transformer for electronic health records, Scientific Reports 10 (2019) 1–12.

[8] I. Landi, B. Glicksberg, H.-C. Lee, S. Cherng, G. Landi, M. Danieletto, C. Furlanello, R. Miotto, Deep representation learning of electronic health records to unlock patient stratification at scale, npj Digital Medicine 3 (2020).

[9] R. Miotto, L. Li, B. A. Kidd, J. T. Dudley, Deep patient: an unsupervised representation to predict the future of patients from the electronic health records, Scientific reports 6 (2016) 1–10.

[10] Y. Huang, N. Wang, Z. Zhang, H. Liu, X. Fei, L. Wei, H. Chen, Patient representation from structured electronic medical records based on embedding technique: Development and validation study, JMIR Medical Informatics 9 (2021).

[11] T. Ruan, L. Lei, Y. Zhou, J. Zhai, L. Zhang, P. He, J. Gao, Representation learning for clinical

time series prediction tasks in electronic health records, BMC Medical Informatics and Decision Making 19-S (2019) 259.

[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] Y. Si, K. Roberts, Patient representation transfer learning from clinical notes based on hierarchical attention network, AMIA Summits on Translational Science Proceedings 2020 (2020) 597.

[15] 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.

[16] 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.

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

[18] 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).

[19] 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.

[20] Centers for Disease Control and Prevention, International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM), https://www.cdc.gov/nchs/icd/icd9cm.htm, 2015. Accessed: 2022-05-01.

[21] 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.

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

[23] 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.

A Galois Framework for the Study of Analogical Classifiers

Miguel Couceiro[1,*], Erkko Lehtonen[2]

[1] LORIA, CNRS, Universite de Lorraine, F-54000, France
[2] Centro de Matemática e Aplicações, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Quinta da Torre, 2829-516 Caparica, Portugal

Abstract

In this paper, we survey some recent advances in the study of analogical classifiers, *i.e.,* classifiers that are compatible with the principle of analogical inference. We will present a Galois framework induced by relation between formal models of analogy and the corresponding classes of analogy preserving functions. The usefulness these general results will be illustrated over Boolean domains, which explicitly present the Galois closed sets of analogical classifiers for different pairs of formal models of Boolean analogies.

Keywords

Analogical proportion, analogical reasoning, analogical classifier, Galois theory

1. Motivation and Background

Analogical reasoning (AR) is a remarkable capability of human thought that exploits parallels between situations of different nature to infer plausible conclusions, by relying simultaneously on similarities and dissimilarities. Machine learning (ML) and artificial intelligence (AI) have tried to develop AR, mostly based on cognitive considerations, and to integrate it in a variety of ML tasks, such as natural language processing (NLP), preference learning and recommendation [1, 2, 3, 4, 5]. Also, analogical extrapolation (inference) can solve difficult reasoning tasks such as *scholastic aptitude tests* and *visual question answering* [6, 7, 8, 9]. Inference based on AR can also support dataset augmentation (analogical extension and extrapolation) for model learning, especially in environments with few labeled examples [10]. Furthermore, AR can also be performed at a meta level for transfer learning [11, 12] where the idea is to take advantage of what has been learned on a source domain in order to improve the learning process in a target domain related to the source domain. Moreover, analogy making can provide useful explanations that rely on the parallel example-counterexample [13] and guide counterfactual generation [14].

However, early works lacked theoretical and formalizational support. The situation started to change about a decade ago when researchers adopted the view of analogical proportions

IARML@IJCAI-ECAI'2022: Workshop on the Interactions between Analogical Reasoning and Machine Learning, at IJCAI-ECAI'2022, July, 2022, Vienna, Austria

[*] Corresponding author.

miguel.couceiro@loria.fr (M. Couceiro); e.lehtonen@fct.unl.pt (E. Lehtonen)
0000-0003-2316-7623 (M. Couceiro); 0000-0002-9255-5876 (E. Lehtonen)

CEUR Workshop Proceedings (CEUR-WS.org)

as statements of the form "a relates to b as c relates to d", usually denoted $a : b :: c : d$. Such proportions are at the root of the analogical inference mechanism, and several formalisms to study this mechanism have been proposed, which follow different axiomatic and logical approaches [15, 16]. For instance, [17] introduces the following 4 postulates in the linguistic context as a guideline for formal models of 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 $d = b$). Such postulates appear reasonable in the word domain, but they can be criticized in other application domains. For instance, in a setting where two distinct conceptual spaces are involved, as in $wine : French :: beer : Belgian$ where two different spaces "drinks" and "nationality" are considered, the central permutation is not tolerable.

Recently, [18] proposed an algebraic framework of analogies that is naturally embedded into first-order logic via model-theoretic types. It provides a unifying setting where the different axiomatic approaches in the literature and respective domains of interpretation can be considered.

Example 1. Among the classical models of analogy on the two-element set $\{0, 1\}$, it is noteworthy to mention [19] and [20] definitions of Boolean analogy that correspond respectively to the relations R_1 and R_2 below. In this paper we represent a relation as a matrix whose columns are precisely the tuples belonging to the relation.

$$R_1 := \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \quad \text{and} \quad R_2 := \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Note that R_2 contains only patterns of the form $x : x :: y : y$ and $x : y :: x : y$, and it is often referred to as the *minimal model* of analogy.

Other approaches to formalizing the notion of analogy, include the *factorial* view of [21] and the *functional* view of [22, 23], except that it is not bound by the central permutation postulate. Such a framework is close to Gentner's symbolic model of analogical reasoning [24] based on *structure mapping theory* and first implemented in [25]. Note that different axiomatic approaches entail different dataset augmentation procedures, and may impact differently on several tasks related to AR.

A key task associated with AR is *analogy solving*, i.e. finding or extrapolating, for a given triple a, b, c a value x such that $a : b :: c : x$ is a valid analogy. Such a task has been addressed in the framework of case-based reasoning (CBR), where solutions are generated by *retrieval* and *adaptation* [26, 27]. Following the same tracks, AR was also adapted to analogy based classification [28] where objects are viewed as attribute tuples (instances) $\mathbf{x} = (x_1, \ldots, x_n)$. Indeed, if $\mathbf{a}, \mathbf{b}, \mathbf{c}$ are in analogical proportion for most of their attributes, and class labels are known for $\mathbf{a}, \mathbf{b}, \mathbf{c}$ but unknown for $\mathbf{d}$, then one may infer the label for $\mathbf{d}$ as a solution of an analogical proportion equation. All these applications rely on the same idea: if four instances $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ are in analogical proportion for most of the attributes describing them, then it may still be the case for the other attributes $f(\mathbf{a}), f(\mathbf{b}), f(\mathbf{c}), f(\mathbf{d})$ (for some function f). This principle is called *analogical inference principle* (AIP).

Theoretically, it is quite challenging to find and characterize situations where AIP can be soundly applied. A first step toward explaining the analogical mechanism consists in characterizing the set of functions f for which AIP is sound (*i.e.*, no error occurs) no matter which triplets of examples are used. In case of Boolean attributes and for the minimal model R_2, it was shown in [10] that these so-called "analogy-preserving" (AP) functions coincide exactly with the set of affine Boolean functions. Moreover, it was also shown that, when the function is not affine, the prediction accuracy remains high if the function is close to being affine [29]. These results were later extended to nominal (finite) underlying sets when taking the minimal model of analogy in both the domain and codomain of classifiers [30].

Intuitively, this class will change when adopting different models of analogy. This motivated a deeper study of the relation between formal models of analogy and the corresponding class of AP functions, and which culminated in a *Galois theory of analogical classifiers* [31]. In this paper, we briefly survey this Galois framework for analogical classifiers, and we illustrate these results for Boolean classifiers by revisiting different formal Boolean models of analogy, and describe the corresponding Galois closed sets of analogical classifiers.

This paper is organized as follows. We first briefly survey the universal algebraic framework pertaining to relational preservation in Section 2. We then adapt the latter to the framework of analogical preservation in Section 3, and recall the Galois theory for analogical classifiers presented in [31]. We illustrate these results by considering two classical models of Boolean analogies and describing sets of analogical classifiers accordingly. As a by-product, it follows that they actually correspond to the set of affine Boolean functions.

2. Galois Theories for Functions

Let A and B be nonempty sets. A *function of several arguments* from A to B is a mapping $f\colon A^n \to B$ for some natural number n called the *arity* of f. Denote by $\mathcal{F}_{AB}^{(n)}$ the set of all n-ary functions of several arguments from A to B, and let $\mathcal{F}_{AB} := \bigcup_{n\in\mathbb{N}} \mathcal{F}_{AB}^{(n)}$.

In the case when $A = B$ we speak of *operations* on A, and we use the notation $\mathcal{O}_A^{(n)} := \mathcal{F}_{AA}^{(n)}$ and $\mathcal{O}_A := \mathcal{F}_{AA}$. For any set $C \subseteq \mathcal{F}_{AB}$, the *$n$-ary part* of C is $C^{(n)} := C \cap \mathcal{F}_{AB}^{(n)}$. If $f \in \mathcal{F}_{BC}^{(n)}$ and $g_1, \dots, g_n \in \mathcal{F}_{AB}^{(m)}$, then the *composition* $f(g_1, \dots, g_n)$ belongs to $\mathcal{F}_{AC}^{(m)}$ and is defined by the rule

$$f(g_1, \dots, g_n)(\mathbf{a}) := f(g_1(\mathbf{a}), \dots, g_n(\mathbf{a})) \quad \text{for all } \mathbf{a} \in A^m.$$

The i-th n-ary *projection* $\mathrm{pr}_i^{(n)} \in \mathcal{O}_A^{(n)}$ is defined by $\mathrm{pr}_i^{(n)}(a_1, \dots, a_n) := a_i$ for all $a_1, \dots, a_n \in A$. We denote by $\mathcal{J}_A$ the set of all projections on A.

The notion of functional composition can be extended to sets of functions as follows. Let $C \subseteq \mathcal{F}_{BC}$ and $K \subseteq \mathcal{F}_{AB}$. The *composition* of C with K is the set

$$CK := \{f(g_1, \dots, g_n) \in \mathcal{F}_{AC} \mid f \in C^{(n)},\, g_1, \dots, g_n \in K^{(m)}\}.$$

A *clone* on A is a set $C \subseteq \mathcal{O}_A$ that is closed under composition and contains all projections, in symbols, $CC \subseteq C$ and $\mathcal{J}_A \subseteq C$. For $F \subseteq \mathcal{O}_A$, we denote by $\langle F \rangle$ the clone generated by F, i.e., the smallest clone on A containing F.

We say that $f \in \mathcal{F}_{AB}^{(n)}$ is a *minor* of $g \in \mathcal{F}_{AB}^{(m)}$, $f \leq g$, if $f \in \{g\}\mathcal{J}_A$. The minor relation $\leq$ is a quasi-order (a reflexive and transitive relation) on $\mathcal{F}_{AB}$. Downsets of $(\mathcal{F}_{AB}, \leq)$ are called *minor-closed* classes or *minions.* Equivalently, a set $C \subseteq \mathcal{F}_{AB}$ is a minion if $C\mathcal{J}_A \subseteq C$. A set $\mathcal{C} \subseteq \mathcal{F}_{AB}$ is m*-locally closed* if for all $f \in \mathcal{F}_{AB}$ (say f is n-ary), it holds that $f \in C$ whenever for every finite subset $S \subseteq A^n$ of size at most m, there exists a $g \in \mathcal{C}$ such that $f|_S = g|_S$. A set $\mathcal{C}$ is said to be *locally closed* if it is m-locally closed for every positive integer m.

Subsets of A^m are called m-ary *relations* on A. Denote by $\mathcal{R}_A^{(m)}$ the set of all m-ary relations on A, and let $\mathcal{R}_A := \bigcup_{m \in \mathbb{N}} \mathcal{R}_A^{(m)}$. Let $f \in \mathcal{O}_A^{(n)}$ and $R \in \mathcal{R}_A^{(m)}$. We say that the function f *preserves* the relation R (or f is a *polymorphism* of R, or R is an *invariant* of f), and we write $f \rhd R$, if for all $\mathbf{a}_1, \dots, \mathbf{a}_n \in R$, we have $f(\mathbf{a}_1, \dots, \mathbf{a}_n) \in R$, where $f(\mathbf{a}_1, \dots, \mathbf{a}_n)$ denotes the componentwise application of f to the tuples $\mathbf{a}_i = (a_{i1}, \dots, a_{im})$, *i.e.:*

$$f(\mathbf{a}_1, \dots, \mathbf{a}_n) := (f(a_{11}, \dots, a_{n1}), \dots, f(a_{1m}, \dots, a_{nm})).$$

The preservation relation $\rhd$ induces a Galois connection between the sets $\mathcal{O}_A$ and $\mathcal{R}_A$ of operations and relations on A. Its polarities are the maps $\operatorname{Pol}\colon \mathcal{P}(\mathcal{R}_A) \to \mathcal{P}(\mathcal{O}_A)$ and $\operatorname{Inv}\colon \mathcal{P}(\mathcal{O}_A) \to \mathcal{P}(\mathcal{R}_A)$ given by the following rules: for all $\mathcal{R} \subseteq \mathcal{R}_A$ and $\mathcal{F} \subseteq \mathcal{O}_A$,

$$\begin{aligned} \operatorname{Pol} \mathcal{R} &:= \{f \in \mathcal{O}_A \mid \forall R \in \mathcal{R}\colon f \rhd R\}, \\ \operatorname{Inv} \mathcal{F} &:= \{R \in \mathcal{R}_A \mid \forall f \in \mathcal{F}\colon f \rhd R\}. \end{aligned}$$

Under this Galois connection, the closed sets of operations are precisely the locally closed clones. The closed sets of relations, known as *relational clones,* are precisely the locally closed sets of relations that contain the empty relation and the diagonal relations and are closed under formation of primitively positively definable relations. This was first shown for finite base sets in [32, 33, 34] and later extended for arbitrary sets in [35, 36].

The preservation relation can be adapted for functions of several arguments from A to B; we now need to consider pairs of relations. Let

$$\mathcal{R}_{AB}^{(m)} := \mathcal{R}_A^{(m)} \times \mathcal{R}_B^{(m)} \quad \text{and} \quad \mathcal{R}_{AB} := \bigcup_{m \in \mathbb{N}} \mathcal{R}_{AB}^{(m)}$$

be the set of all (m-ary) *relational constraints* from A to B.

Let $f \in \mathcal{F}_{AB}^{(n)}$ and $(R, S) \in \mathcal{R}_{AB}^{(m)}$. We say that f *preserves* (R, S) (or f is a *polymorphism* of (R, S), or (R, S) is an *invariant* of f), and we write $f \rhd (R, S)$, if for all $\mathbf{a}_1, \dots, \mathbf{a}_n \in R$, we have $f(\mathbf{a}_1, \dots, \mathbf{a}_n) \in S$. As before, the preservation relation $\rhd$ induces a Galois connection between the sets $\mathcal{F}_{AB}$ and $\mathcal{R}_{AB}$ of functions and relational constraints from A to B. Its polarities are the maps $\operatorname{Pol}\colon \mathcal{P}(\mathcal{R}_{AB}) \to \mathcal{P}(\mathcal{F}_{AB})$ and $\operatorname{Inv}\colon \mathcal{P}(\mathcal{F}_{AB}) \to \mathcal{P}(\mathcal{R}_{AB})$ given by the following rules: for all $\mathcal{Q} \subseteq \mathcal{R}_{AB}$ and $\mathcal{F} \subseteq \mathcal{F}_{AB}$,

$$\begin{aligned} \operatorname{Pol} \mathcal{Q} &:= \{f \in \mathcal{F}_{AB} \mid \forall (R, S) \in \mathcal{Q}\colon f \rhd (R, S)\}, \\ \operatorname{Inv} \mathcal{F} &:= \{(R, S) \in \mathcal{R}_{AB} \mid \forall f \in \mathcal{F}\colon f \rhd (R, S)\}. \end{aligned}$$

The sets $\operatorname{Pol} \mathcal{Q}$ and $\operatorname{Inv} \mathcal{F}$ are said to be *defined* by $\mathcal{Q}$ and $\mathcal{F}$, respectively. Sets of functions of the form $\operatorname{Pol} \mathcal{Q}$ for some $\mathcal{Q} \subseteq \mathcal{R}_{AB}$ and sets of relational constraints of the form $\operatorname{Inv} \mathcal{F}$ for some $\mathcal{F} \subseteq \mathcal{F}_{AB}$ are said to be *definable* by relational constraints and functions, respectively.

The closed sets of functions under this Galois connection were described for finite base sets in [37] and later for arbitrary sets [38]. This result was refined in [39] for sets of functions definable by relations of restricted arity.

Theorem 2 ([38, 39])**.** *Let A and B be arbitrary nonempty sets, and let $C \subseteq \mathcal{F}_{AB}$.*

1. *C is definable by constraints if and only if C is a locally closed minion.*
2. *C is definable by constraints of arity m if and only if C is an m-locally closed minion.*

The closed sets of relational constraints were described in terms of closure conditions that parallel those for relational clones. The description of the dual objects of constraints on possibly infinite sets A and B was also provided in [38] and inspired by those given in [34, 35, 36, 37] and given in terms of positive primitive first-order relational definitions applied simultaneously on antecedents and consequents. Sets of constraints that are closed under such formation schemes are said to be *closed under conjunctive minors.* Moreover, every function satisfies the empty $(\varnothing, \varnothing)$ and the equality $(=_A, =_B)$ constraints, and if a function f satisfies a constraint (R, S), then f also satisfies its *relaxations* (R', S') such that $R' \subseteq R$ and $S' \supseteq S$.

As for functions, in the infinite case, we also need to consider a "local closure" condition to describe the dual closed sets of relational constraints on A and B. A set $\mathcal{Q}$ of constraints on A and B is n*-locally closed* if it contains every relaxation of its members whose antecedent has size at most n, and it is *locally closed* if it is n-locally closed for every positive integer n.

Theorem 3 ([38, 39])**.** *For arbitrary nonempty sets A and B, and let $\mathcal{Q} \subseteq \mathcal{R}_{AB}$ be a set of relational constraints on A and B.*

1. *$\mathcal{Q}$ is definable by some set $\mathcal{C} \subseteq \mathcal{F}_{AB}$ if and only if it is locally closed, contains the binary equality and the empty constraints, and it is closed under relaxations and conjunctive minors.*
2. *$\mathcal{Q}$ is definable by some set $\mathcal{C} \subseteq \mathcal{F}_{AB}^{(n)}$ of n-ary functions if and only it is n-locally closed, contains the binary equality and the empty constraints, and it is closed under relaxations and conjunctive minors.*

Let $K \subseteq \mathcal{F}_{AB}$ and let C_1 and C_2 be clones on A and B. We say that K is *stable under right composition with* C_1 if $KC_1 \subseteq K$, and we say that K is *stable under left composition with* C_2 if $C_2K \subseteq K$. We say that K is (C_1, C_2)*-stable* or a (C_1, C_2)*-clonoid,* if $KC_1 \subseteq K$ and $C_2K \subseteq K$.

Motivated by earlier results on linear definability of equational classes of Boolean functions [40] which were described in terms of stability under compositions with the clone of constant preserving affine functions, [41] introduced a Galois framework for describing sets of functions $\mathcal{F} \subseteq \mathcal{F}_{AB}$ stable under right and left compositions with clones C_1 on A and C_2 on B, respectively. For that they restricted the defining dual objects to relational constraints (R, S) where R and S invariant under C_1 and C_2, respectively, i.e., $R \in \operatorname{Inv} C_1$ and $S \in \operatorname{Inv} C_2$. These were referred to as (C_1, C_2)*-constraints.* We denote by $\mathcal{R}_{AB}^{(C_1,C_2)}$ the set of all (C_1, C_2)-constraints.

Theorem 4 ([41])**.** *Let A and B be arbitrary nonempty sets, and let C_1 and C_2 clones on A and B, respectively. A set $\mathcal{C} \subseteq \mathcal{F}_{AB}$ is definable by some set of (C_1, C_2)-constraints if and only if $\mathcal{C}$ is locally closed and stable under right and left composition with C_1 and C_2, respectively, i.e., it is a locally closed (C_1, C_2)-clonoid.*

Dually, a set $\mathcal{Q}$ of (C_1, C_2)-constraints is definable by a set $\mathcal{C} \subseteq \mathcal{F}_{AB}$ if $\mathcal{Q} = \operatorname{Inv} \mathcal{C} \cap \mathcal{R}_{AB}^{(C_1,C_2)}$. To describe the dual closed sets of (C_1, C_2)-constraints, [41] observed that conjunctive minors of (C_1, C_2)-constraints are themselves (C_1, C_2)-constraints. However, this is not the case for relaxations. They thus proposed the following variants of local closure and of constraint relaxations.

A set $\mathcal{Q}_0$ of (C_1, C_2)-constraints is said to be (C_1, C_2)-*locally closed* if the set $\mathcal{Q}$ of all relaxations of the various constraints in $\mathcal{Q}_0$ is locally closed. A relaxation (R_0, S_0) of a relational constraint (R, S) is said to be a (C_1, C_2)-*relaxation* if (R_0, S_0) is a (C_1, C_2)-constraint.

Theorem 5 ([41])**.** *Let A and B be arbitrary nonempty sets, and let C_1 and C_2 clones on A and B, respectively. A set $\mathcal{Q}$ of (C_1, C_2)-constraints is definable by some set $\mathcal{C} \subseteq \mathcal{F}_{AB}$ if and only if it is (C_1, C_2)-locally closed and contains the binary equality constraint, the empty constraint, and it is closed under (C_1, C_2)-relaxations and conjunctive minors.*

In this paper, we will focus on relational constraints whose antecedent and consequent are derived from analogies, and that we will refer to as *analogical constraints*. We will denote the set of all analogical constraints from A to B by $\mathcal{A}_{AB}$.

3. Galois Theory for Analogical Classifiers

As mentioned in the Introduction, analogical inference yields competitive results in classification and recommendation tasks. However, the justification of why and when a classifier is compatible with the analogical inference principle (AIP) remained rather obscure until the work [10]. In this paper the authors considered the minimal Boolean analogy model (see R_2 in Example 1) and addressed the problem of determining those *Boolean classifiers for which the AIP always holds*, that is, for which there are no classification errors. Surprisingly, they showed that they correspond to "analogy preserving" and that they constitute the clone of affine functions. This result was later generalized to binary classification tasks on nominal (finite) domains in [30] where the authors considered the more stringent notion of "hard analogy preservation". By taking the same minimal analogy model on both the domain and the label set, the authors showed that in this case the sets of hard analogy preserving functions constitute Burle's clones [42].

Definition 6. Let A and B be sets, and let R and S be analogical proportions defined on the two sets, respectively. A function $f\colon A^n \to B$ is *analogy-preserving* (AP for short) relative to (R, S) if for all $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in A^n$:

$$\big(R(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) \quad \text{and} \quad S\text{-solv}(f(\mathbf{a}), f(\mathbf{b}), f(\mathbf{c}))\big) \quad \Longrightarrow \quad S(f(\mathbf{a}), f(\mathbf{b}), f(\mathbf{c}), f(\mathbf{d})),$$

where $R(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$ is a shorthand for $(a_i, b_i, c_i, d_i) \in R$ for all $i \in \{1, \dots, n\}$ and S-solv$(f(\mathbf{a}), f(\mathbf{b}), f(\mathbf{c}))$ means that there is an $x \in B$ with $S(f(\mathbf{a}), f(\mathbf{b}), f(\mathbf{c}), x)$. Denote by $\mathsf{AP}(R, S)$ the set of all analogy-preserving functions relative to (R, S).

This relation between functions and formal models of analogy gives rise to a Galois connection whose closed sets of functions correspond exactly to the classes of analogical classifiers. As the following result shows, we can use the universal algebraic tools of Section 2 to investigate analogy preservation.

Proposition 7. *Let R and S be analogical proportions defined on sets A and B, respectively. Then* $\mathsf{AP}(R, S) = \mathrm{Pol}(R, S')$*, where*

$$S' := S \cup \{(a, b, c, d) \in B^4 \mid \nexists x \in B\colon (a, b, c, x) \in S\}. \tag{1}$$

Consequently, $\mathsf{AP}(R, S)$ *is a locally closed minion.*

Example 8. The derived relations as in Proposition 7 corresponding to the formal models of Boolean analogies in Example 1 are the following:

$$R_1' = R_1 \quad \text{and} \quad R_2' = R_2 \cup \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

To fully describe the sets of the form $\mathrm{Pol}(R, S')$ we need to introduce some variants of the closure conditions discussed in Section 2. Let $\mathcal{R}$ be set of m-ary relations on A. An $m \times n$ matrix D whose columns belong to a relation $R \in \mathcal{R}$, is called an *$\mathcal{R}$-locality*. Let $\mathcal{Q} \subseteq \mathcal{R}_{AB}$, and let $\mathcal{Q}_1 := \{R \in \mathcal{R}_A \mid \exists S \in \mathcal{R}_B$ such that $(R, S) \in \mathcal{Q}\}$. A set $\mathcal{C} \subseteq \mathcal{F}_{AB}$ is *Q-locally closed* if for all $f \in \mathcal{F}_{AB}$ (say f is n-ary), it holds that $f \in C$ whenever for every $\mathcal{Q}_1$-locality D, either

1. there exists a $g \in \mathcal{C}$ such that $fD = gD$, or
2. for any relation R in $\mathcal{Q}_1$ such that $D \preceq R$ and for any

$$T \in \{S \in \mathcal{R}_B \mid (R, S) \in \mathcal{Q},\, \mathcal{C}R \subseteq S\}$$

we have that $fR \subseteq T$.

Let $\mathcal{A}_B' := \{S' \mid S \in \mathcal{A}_B\}$, and let $\mathcal{A}_{AB} := \mathcal{A}_A \times \mathcal{A}_B'$. We refer to the elements of $\mathcal{A}_{AB}$ as *analogical constraints* from A to B. The set of analogical constraints that are (C_1, C_2)-constraints will be denoted by

$$\mathcal{A}_{AB}^{(C_1, C_2)} := \mathcal{A}_{AB} \cap \mathcal{R}_{AB}^{(C_1, C_2)}.$$

A set $\mathcal{C}$ is said to be *(C_1, C_2)-analogically locally closed* if it is $\mathcal{A}_{AB}^{(C_1, C_2)}$-locally closed. Note that $\mathcal{A}_{AB} = \mathcal{A}_{AB}^{(\mathcal{J}_A, \mathcal{J}_B)}$, and in this case we simply say that $\mathcal{C}$ is *analogically locally closed.*

Theorem 9. *Let A and B be arbitrary nonempty sets, and let C_1 and C_2 be clones on A and B, respectively.*

1. *A set $\mathcal{C} \subseteq \mathcal{F}_{AB}$ is definable by analogical (C_1, C_2)-constraints if and only if it is a (C_1, C_2)-analogically locally closed (C_1, C_2)-clonoid.*
2. *A set $\mathcal{C} \subseteq \mathcal{F}_{AB}$ is definable by analogical constraints if and only if it is an analogically locally closed minion.*

4. Application: Description of Boolean Analogical Classifiers w.r.t. Example 1

In this section we illustrate the use of the Galois theory described in Section 3 to determine the classes of analogical classifiers $\mathsf{AP}(R_i, R_j) = \mathrm{Pol}(R_i, R'_j)$ for $i, j \in \{1, 2\}$ (see Example 8 and Equation (1)).

Recall that, up to permutation of arguments, the binary Boolean functions are the following: the constant 0 and 1 functions, denoted respectively by 0 and 1, the first projection $\mathrm{pr}_1 \colon (x_1, x_2) \mapsto x_1$ and its negation $\neg_1 = \overline{\mathrm{pr}_1}$, the conjunction $\wedge$ and its negation $\uparrow$, the disjunction $\vee$ and its negation $\downarrow$, the implication $\rightarrow$ and its negation $\not\rightarrow$, and the addition $+$ modulo 2 and its negation $\leftrightarrow$. Note that $\uparrow$ and $\downarrow$ are often referred to as *Sheffer functions* as each one of them can generate the class of all Boolean functions by taking compositions and variable substitutions.

Observe that the constant tuples $\mathbf{0}$ and $\mathbf{1}$ belong to every R_i ($i \in \{1, 2\}$), and thus every such R_i is invariant under I, i.e., $\mathsf{I}R_i \subseteq R_i$. Hence, for every $i, j \in \{1, 2\}$, $\mathrm{Pol}(R_i, R'_j)$ is stable under right composition with I. This leads us to considering the following notion.

A function f is said to be a *C-minor* of a function g if $f \in gC$. Recall that in the particular case when $C = \mathcal{J}_{\{0,1\}}$, f is called a minor of g. The functions f and g are said to be *equivalent*, denoted by $f \equiv g$, if f is a minor of g and g is a minor of f. For further background on these notions and variants see, e.g., [43, 44, 37].

Since $\mathrm{Pol}(R_i, R'_j)$ is stable under right composition with I, this means that if an I-minor f of a function g does not belong to $\mathrm{Pol}(R_i, R'_j)$, then neither does g. This observation constitutes a main tool in describing the sets of the form $\mathsf{AP}(R_i, R_j) = \mathrm{Pol}(R_i, R'_j)$.

Proposition 10. *Let* L *denote the clone of affine functions. We have* $\mathsf{AP}(R_2, R_2) = \mathsf{AP}(R_2, R_1) = \mathsf{AP}(R_1, R_2) = \mathsf{AP}(R_1, R_1) = \mathsf{L}$.

Proof. We make use of the fact that $\mathsf{AP}(R, S) = \mathrm{Pol}(R, S')$. Since $R_1 = R'_1$, it follows immediately that $\mathrm{Pol}(R_1, R'_1) = \mathrm{Pol}\, R_1$ is a clone, and it is well known that $\mathrm{Pol}\, R_1 = \mathsf{L}$.

To prove $\mathsf{AP}(R_2, R_2) = \mathrm{Pol}(R_2, R'_2) = \mathsf{L}$, we make use of main tool given above. Since (R_2, R'_2) is a relaxation of (R_1, R'_1), $\mathrm{Pol}(R_2, R'_2) \supseteq \mathrm{Pol}(R_1, R'_1) = \mathsf{L}$. Furthermore,

- $\wedge, \vee \notin \mathrm{Pol}(R_2, R'_2)$ because

$$\wedge \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \notin R'_2 \qquad \text{and} \qquad \vee \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \notin R'_2.$$

- $\uparrow, \downarrow \notin \mathrm{Pol}(R_2, R'_2)$ because

$$\uparrow \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \notin R'_2, \qquad \text{and} \qquad \downarrow \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \notin R'_2.$$

- $\nrightarrow, \rightarrow \notin \mathrm{Pol}(R_2, R_2')$ because

$$\nrightarrow \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \notin R_2', \qquad \text{and} \qquad \rightarrow \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \notin R_2'.$$

The result now follows by observing that any function outside of L has an I-minor in $\{\wedge, \vee, \uparrow, \downarrow, \nrightarrow, \rightarrow\}$. (For further details, see [31].) By similar arguments, it also follows that $\mathsf{AP}(R_2, R_1) = \mathsf{AP}(R_1, R_2) = \mathsf{AP}(R_1, R_1) = \mathsf{L}$. □

5. Conclusion and Perspectives

In this paper we survey a general Galois framework for studying analogical classifiers that does not depend on the underlying domains nor the formal models of analogy considered. We also illustrate its usefulness by explicitly describing sets of analogical classifiers with respect to two classical models of analogy. As future work, we intend to further explore different formal models of analogy that may be obtained by considering different algebraic signatures.

Acknowledgments

The authors wish to thank Esteban Marquer, Pierre-Alexandre Murena and the reviewers for the useful suggestions for improving this manuscript.

The research work by the first named author 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).

The research work by the second named author was partially funded by national funds through the FCT – Fundação para a Ciência e a Tecnologia, I.P., under the scope of the projects PTDC/MAT-PUR/31174/2017, UIDB/00297/2020, and UIDP/00297/2020 (Center for Mathematics and Applications).

References

[1] 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.

[2] M. A. Fahandar, E. Hüllermeier, Learning to rank based on analogical reasoning, in: AAAI-18, 2018, pp. 2951–2958.

[3] M. A. Fahandar, E. Hüllermeier, Analogical embedding for analogy-based learning to rank, in: IDA 2021, volume 12695 of *LNCS*, Springer, 2021, pp. 76–88.

[4] N. Hug, H. Prade, G. R., M. Serrurier, Analogical proportion-based methods for recommendation – first investigations, Fuzzy Sets Syst. 366 (2019) 110–132.

[5] M. Mitchell, Abstraction and analogy-making in artificial intelligence, 2021. URL: https://arxiv.org/abs/2102.10717.

[6] P. D. Turney, The latent relation mapping engine: Algorithm and experiments, J. Artif. Intell. Res. 33 (2008) 615–655.

[7] P. D. Turney, P. Pantel, From frequency to meaning: Vector space models of semantics, J. Artif. Intell. Res. 37 (2010) 141–188.

[8] F. Sadeghi, C. L. Zitnick, A. Farhadi, Visalogy: Answering visual analogy questions, in: NIPS 2015, 2015, pp. 1882–1890.

[9] J. Peyre, I. Laptev, C. Schmid, J. Sivic, Detecting unseen visual relations using analogies, in: IEEE ICCV 2019, 2019, pp. 1981–1990.

[10] M. Couceiro, N. Hug, H. Prade, G. Richard, Analogy-preserving functions: A way to extend Boolean samples, in: IJCAI, ijcai.org, 2017, pp. 1575–1581.

[11] A. Cornuéjols, P.-A. Murena, R. Olivier, Transfer learning by learning projections from target to source, in: IDA 2020, volume 12080 of *LNCS*, Springer, 2020, pp. 119–131.

[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, 2021, pp. 76–89.

[13] E. Hüllermeier, Towards analogy-based explanations in machine learning, in: MDAI 2020, volume 12256 of *LNCS*, Springer, 2020, pp. 205–217.

[14] M. T. Keane, B. Smyth, Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI), in: ICCBR 2020, volume 12311 of *LNCS*, Springer, 2020, pp. 163–178.

[15] Y. Lepage, Analogy and formal languages, Electron. Notes Theor. Comput. Sci. 53 (2001) 180–191.

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

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

[18] C. Antić, Analogical proportions, 2020. URL: https://arxiv.org/abs/2006.02854.

[19] S. Klein, Culture, mysticism & social structure and the calculation of behavior, in: Proc. 5th Europ. Conf. in Artificial Intelligence (ECAI'82), 1982, pp. 141–146.

[20] L. Miclet, H. Prade, Handling analogical proportions in classical logic and fuzzy logics settings, in: ECSQARU, Springer, 2009, pp. 638–650.

[21] N. Stroppa, F. Yvon, An analogical learner for morphological analysis, in: CoNLL 2005, ACL, 2005, pp. 120–127.

[22] N. Barbot, L. Miclet, H. Prade, Analogy between concepts, Artif. Intell. 275 (2019) 487–539.

[23] P.-A. Murena, M. Al-Ghossein, J.-L. Dessalles, A. Cornuéjols, Solving analogies on words based on minimal complexity transformation., in: IJCAI, 2020, pp. 1848–1854.

[24] D. Gentner, C. Hoyos, Analogy and abstraction, Top. Cogn. Sci. 9 (2017) 672–693.

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

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

[27] J. Lieber, E. Nauer, H. Prade, When revision-based case adaptation meets analogical extrapolation, in: ICCBR 2021, volume 12877 of *LNCS*, Springer, 2021, pp. 156–170.

[28] M. Bounhas, H. Prade, G. Richard, Analogy-based classifiers for nominal or numerical data, IJAR 91 (2017) 36–55.

[29] M. Couceiro, N. Hug, H. Prade, G. Richard, Behavior of analogical inference w.r.t. Boolean

functions, in: IJCAI, ijcai.org, 2018, pp. 2057–2063.

[30] M. Couceiro, E. Lehtonen, L. Miclet, H. Prade, G. Richard, When nominal analogical proportions do not fail, in: SUM 2020,, volume 12322 of *LNCS*, Springer, 2020, pp. 68–83.

[31] M. Couceiro, E. Lehtonen, Galois theory for analogical classifiers, CoRR abs/2205.04593 (2022). URL: https://doi.org/10.48550/arXiv.2205.04593.

[32] V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, B. A. Romov, Galois theory for Post algebras I, Kibernetika 5 (1969) 1–10.

[33] V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, B. A. Romov, Galois theory for Post algebras II, Kibernetika 5 (1969) 1–9.

[34] D. Geiger, Closed systems of functions and predicates, Pacific J. Math. 27 (1968) 95–100.

[35] L. Szabó, Concrete representation of related structures of universal algebras I, Acta Sci. Math. (Szeged) 40 (1978) 175–184.

[36] R. Pöschel, Concrete representation of algebraic structures and a general Galois theory, in: H. Kautschitsch, W. B. Müller, W. Nöbauer (Eds.), Contributions to General Algebra (Proc. Klagenfurt Conf., Klagenfurt, 1978), Johannes Heyn, Klagenfurt, 1979, pp. 249–272.

[37] N. Pippenger, Galois theory for minors of finite functions, Discrete Math. 254 (2002) 405–419.

[38] M. Couceiro, S. Foldes, On closed sets of relational constraints and classes of functions closed under variable substitutions, Algebra Universalis 54 (2005) 149–165.

[39] M. Couceiro, On Galois connections between external operations and relational constraints: arity restrictions and operator decompositions, Acta Sci. Math. 72 (2006) 15–35.

[40] M. Couceiro, S. Foldes, Definability of Boolean function classes by linear equations over $\mathbf{GF}(2)$, Discrete Appl. Math. 142 (2004) 29–34.

[41] M. Couceiro, S. Foldes, Function classes and relational constraints stable under compositions with clones, Discuss. Math., Gen. Algebra Appl. 29 (2009) 109–121.

[42] G. A. Burle, Classes of k-valued logic which contain all functions of a single variable, Diskret. Analiz, Novosibirsk 10 (1967) 3–7.

[43] E. Lehtonen, Descending chains and antichains of the unary, linear, and monotone subfunction relations, Order 23 (2006) 129–142.

[44] E. Lehtonen, Á. Szendrei, Partial orders induced by quasilinear clones, in: Contributions to General Algebra 20, Verlag Johannes Heyn, Klagenfurt, 2012, pp. 51–84.

Measuring the Feasibility of Analogical Transfer using Complexity

Pierre-Alexandre Murena[1,†]

[1]Helsinki Institute for Information Technology HIIT, Department of Computer Science, Aalto University, Espoo, Finland

Abstract

Analogies are 4-ary relations of the form “A is to B as C is to D". While focus has been mostly on how to solve an analogy, i.e. how to find correct values of D given A, B and C, less attention has been drawn on whether solving such an analogy was actually feasible. In this paper, we propose a quantification of the transferability of a source case (A and B) to solve a target problem C. This quantification is based on a complexity minimization principle which has been demonstrated to be efficient for solving analogies. We illustrate these notions on morphological analogies and show its connections with machine learning, and in particular with Unsupervised Domain Adaptation.

Keywords

Analogical reasoning, Analogical transfer, Minimum Message Length, Domain adaptation

1. Introduction

Analogies are 4-ary relations of the form “A is to B as C is to D", denoted $A : B :: C : D$. Even though humans demonstrate strong capabilities of understanding and generating analogies, which has been intensively studied by cognitive sciences [1], these tasks are much more difficult for a machine. In particular, an important task consists in solving analogical equations: given A, B and C, find D such that $A : B :: C : D$ is a valid analogy. Solving such equations has been investigated in multiple domains: Boolean domains [2], formal concepts [3], structured character strings [4], semantic [5, 6] or morphological tasks [7].

In most cases, the aforementioned methods are designed to provide an answer to any equation $A : B :: C : x$, regardless of whether the equation makes sense. This is not the case for humans, who consider that some analogies make more sense than others. Consider for instance the domain of Hofstadter analogies [4]. It describes analogies between character strings, with strong domain constraints relative to the order of the alphabet. For instance, the analogy “ABC : ABD :: IJK : IJL", which is a typical illustrative example of this domain, is based on the notions of *increment* and *last element*. Intuitively, not all analogical equations have a solution in this domain: for instance, it is difficult for a human to find a satisfying solution to the equation “ABC : HIC :: BFQ : x". This is confirmed by the results of the user study conducted by Murena et al. [8].

IARML@IJCAI-ECAI'2022: Workshop on the Interactions between Analogical Reasoning and Machine Learning, at IJCAI-ECAI'2022, July, 2022, Vienna, Austria

pierre-alexandre.murena@aalto.fi (P. Murena)
http://pamurena.com/ (P. Murena)
0000-0003-4586-9511 (P. Murena)

CEUR Workshop Proceedings (CEUR-WS.org)

Which analogical equations are indeed solvable, is a rather infrequently discussed question. However, it would have strong implications, be it in practical applications of analogies (e.g. in intelligent tutoring systems [9]) or in transfer learning. In this paper, we propose a first step toward this important direction, by considering how to measure the transferability of a source case (A, B) to a target case (C, D). To do so, we propose a very general formalization of the problem, which is shown to apply to transfer in both symbolic tasks (like morphological analogies) and numerical machine learning. Based on this formalization, and getting inspiration from applications of Kolmogorov complexity in inference [10], we propose several potential definitions of transferability and discuss their main properties.

2. Preliminary Notions

2.1. Domains and Model Spaces

Although various definitions of analogy have been proposed in the literature, we consider in this paper the following definitions, inspired by the recent framework of Antic [11].

A domain $\mathcal{D}$ is defined as the product of two spaces $\mathcal{X}$ (the *problem space*) and $\mathcal{Y}$ (the *solution space*). An element $(x, y) \in \mathcal{D}$ will be referred to as a *case*.

We introduce a set $\mathcal{R}$ called *representation space*. A model space $\mathbb{M}_{\mathcal{R},\mathcal{D}}$ of domain $\mathcal{D}$ based on representation $\mathcal{R}$ is defined as a subset $\mathbb{M}_{\mathcal{R},\mathcal{D}} \subseteq \{f : \mathcal{X} \times \mathcal{Y} \times \mathcal{R} \to [0, 1]\}$ of functions mapping a problem $x \in \mathcal{X}$, a solution $y \in \mathcal{Y}$ and a representation $r \in \mathcal{R}$, to a real number. Any $M \in \mathbb{M}_{\mathcal{D}}$ is called a model of $\mathcal{D}$. When the context is clear, we will use the notation $\mathbb{M}$ instead of $\mathbb{M}_{\mathcal{R},\mathcal{D}}$.

We illustrate these notions with two examples that will be further investigated in Sections 4 and 5.

Example 1 (Recursive model for morphology)**.** *Given an alphabet $\mathcal{A}$, we define by $\mathcal{A}^*$ the set of words of $\mathcal{A}$. The morphological domain consists of two forms of a word (e.g. declension of a word or conjugation of a verb in natural language): therefore, it is given by $\mathcal{D} = \mathcal{A}^* \times \mathcal{A}^*$.*

We define the representation space $\mathcal{R} = \bigcup_{n=1}^{\infty} (\mathcal{A}^*)^n$. Let $\mathcal{F}$ be the space of all recursive functions [1] from $\mathcal{R}$ to $\mathcal{D}$. For all $\phi \in \mathcal{F}$, we define $M_\phi : \mathcal{A}^* \times \mathcal{A}^* \times \mathcal{R} \to \{0, 1\}$ as:

$$M_\phi(x, y, r) = \begin{cases} 1 & \textit{if } \phi(r) = (x, y) \\ 0 & \textit{otherwise} \end{cases} \tag{1}$$

We can then define the model space $\mathbb{M}_{\mathcal{R},\mathcal{D}}$ as:

$$\mathbb{M}_{\mathcal{R},\mathcal{D}} = \{M_\phi | \phi \in \mathcal{F}\} \tag{2}$$

This example has an easy interpretation. The morphological domain consists of two words w_1 and w_2, which are typically two flections of a same word. For instance, the tuple “play : played" describes the flection of the English verb “play" from the present tense to the perfect tense. Similarly, the tuple “taloon : talossa" describes the flection of the Finnish noun “talo" from the illative case to the inessive case.

[1] i.e. functions computable by a Turing machine. See Section 2.2.

For a better readability, we decompose the recursive function $\phi \in \mathcal{F}$ as $\phi(r) = (\phi_1(r), \phi_2(r))$. The function ϕ_1 (resp. ϕ_2) describes how the word w_1 (resp. w_2) is formed based on some representation r. For instance, in the "play : played" example, we can have $\phi_1(r) = r$, $\phi_2(r) = r +$ "ed" (where the $+$ operation is the string concatenation), and both functions are instantiated with $r =$ "play".

Example 2 (Probabilistic models on $\mathbb{R}^d$)**.** *We define the binary domain $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ with problem space $\mathcal{X} = (\mathbb{R}^d)^*$ and solution space $\mathcal{Y} = \mathcal{L}^*$, for some space $\mathcal{L}$ of labels. When $\mathcal{L}$ is discrete, the problem is called* classification, *otherwise* regression. *An observation on domain $\mathcal{D}$ consists of one labelled dataset, where the problem is the unlabeled dataset (points in $\mathbb{R}^d$) and the labels (in $\mathcal{L}$).*

Let $\mathcal{P}$ be a set $\mathcal{P}_\Theta = \{p_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow [0,1] | \theta \in \Theta\}$ of probability density functions on $\mathcal{D}$ parameterized by $\theta \in \Theta$. Fixing $\mathcal{R} = \emptyset$ and therefore identifying $M(x, y, r)$ to $f(x, y)$, we can define the model space $\mathbb{M}_{\mathcal{R},\mathcal{D}} = \mathcal{P}_\Theta$.

In this paper, we assume that all considered quantities are computable. For those which are not computable (for instance domains of real numbers used in Example 1), we will introduce relevant computable approximations.

2.2. Kolmogorov Complexity

Our framework relies on the use of Kolmogorov complexity [10]. We propose a gentle introduction to this notion. In the following, we will use the notation $\mathbb{B}$ to designate the binary set $\{0, 1\}$.

A function $\phi : \mathbb{B}^* \rightarrow \mathbb{B}^*$ is called partial recursive if its output $\phi(p)$ corresponds to the output of a given Turing machine after its execution with input p when it halts (otherwise, we use the convention $\phi(p) = \infty$). With this notation, the function ϕ can be improperly likened to a Turing machine. In this case, the input p is called a *program*. A partial recursive function ϕ is called *prefix* if, for all $p, q \in \mathbb{B}^*$, if $\phi(p) < \infty$ and $\phi(q) < \infty$, then p is not a proper prefix of q.

Complexity $K_\phi(x)$ of a string $x \in \mathbb{B}^*$, relative to a partial recursive prefix (p.r.p.) function ϕ, is defined as the length of the shortest string p such that $\phi(p) = x$:

$$K_\phi(x) = \min_{p \in \mathbb{B}^*} \{l(p) : \phi(p) = x\} \tag{3}$$

where $l(p)$ represents the length of the string p.

A key result of the theory of complexity is the existence of an additively optimal p.r.p. function ϕ_0: for any p.r.p. function ϕ, there exists a constant c_ϕ such that for all $x \in \mathbb{B}^*$, $K_{\phi_0}(x) \leq K_\phi(x) + c_\phi$. These additively optimal p.r.p. functions are used to define Kolmogorov complexity. They present in particular invariance properties, which means that the difference between the complexities defined by two distinct universal p.r.p. functions is bounded. However, it can be shown that Kolmogorov complexity is not computable, and thus cannot be used in practice.

In practice, this limitation is overcome by fixing a reference p.r.p. function which is not optimal but leads to a computable complexity. By definition, this non-optimal complexity is an upper-bound of Kolmogorov complexity (up to an additive constant). This choice of a reference function is particularly restrictive and imposes some biases, which is inherent to any inductive problem. In the following, we will refer to this upper-bound either as complexity or as description length.

2.3. Inference on a Single Domain

Consider a domain $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ and an observation $(x, y) \in \mathcal{D}$. Given a model space $\mathbb{M}_\mathcal{D}$, the standard *inference* task of supervised learning is to select a model $M \in \mathbb{M}_\mathcal{D}$ which optimally describes the observation.

Selecting a model which accurately describes the observation is not enough in general: In most situations there exists multiple such models, and sometimes even infinitely-many. It is then necessary to discriminate among all these models, in particular based on the intended use of the model. In statistical learning for instance, the discrimination is done by both the selection of a *simple* model family and a *penalization* of models [12].

The inference of the model describing an observed case can then be split into two aspects: selection of a model accurately describing the case and penalization over the model space. This idea has been formalized by Algorithmic Information Theory using Kolmogorov complexity. The *Minimum Message Length* (MML) [13] and the *crude Minimum Description Length* (MDL) [14] principles both formulate the general inference task over a domain as the following minimization problem:

$$\underset{M \in \mathbb{M}_\mathcal{D}}{\text{minimize}} \qquad K(M) + K(x, y|M) \tag{4}$$

This two-part objective function corresponds to the trade-off between the accuracy of the model and its simplicity. For instance, in the morphological domain, it is possible to define a model accounting for all possible valid transformations. By construction, this model will have perfect accuracy, but it will require to encode every pair of problems and solutions in the language, and therefore will be particularly complex.

We remind the computation of complexity is relative to a choice of a reference Turing machine. Therefore, this choice imposes a strong bias over the intended outcome. A *refined* version of the MDL principle has been proposed, which overcomes this limitation. This version is beyond the scope of this paper, but we refer the interested reader to (Grünwald, 2007) [15].

3. A Definition of Transferability

Analogies involve two separate domains: a *source domain* $\mathcal{D}^S$ and a *target domain* $\mathcal{D}^T$. Given a source problem-solution pair $(x^S, y^S) \in \mathcal{D}^S$ and a target problem $x^T \in \mathcal{X}^T$, the analogical transfer from (x^S, y^S) to x^T is informally defined as finding $y^T \in \mathcal{Y}^T$ such that the transformation $x^S \mapsto y^S$ is “similar" to the transformation $x^T \mapsto y^T$. This notion of similarity is problematic, especially when the source and target domains are distinct. Existing frameworks of analogy often assume that the source and target domains are the same [2], or at least share a common structure (e.g. are L-algebras of a same language L, such as proposed by (Antic, 2020) [11]).

In this section, we show how complexity can be used to properly define this similarity, even in the case of distinct domains. We will then show that this definition helps quantifying the notion of *transferability*, i.e. how the source observation (x^S, y^S) is useful to find a solution to target problem x^T.

3.1. Inference of a Target Model

The task in the target domain consists in predicting the solution $y^T \in \mathcal{Y}^T$ associated to the problem $x^T \in \mathcal{X}^T$. Usually, this is done throughout a model M, in particular by finding $y^T \in \mathcal{Y}^T$ and $r \in \mathcal{R}$ maximizing the score $M(x^T, y^T, r)$:

$$y^{T*} \in \arg\max_{y \in \mathcal{Y}^T} \left\{ \max_{r \in \mathcal{R}} M(x^T, y, r) \right\} \tag{5}$$

In the context of Example 1, this corresponds to choosing a representation r that successfully describes x^T given the recursive function ϕ associated to M, i.e. finding r such that $\phi_1(r) = x^T$. The solution y^T is then estimated by taking $y^T = \phi_2(r)$. In the context of Example 2, Equation (5) corresponds to taking the most probable solution.

However, in practice, the model M is not known and needs to be inferred. The difficulty is that, in general, it is not possible to identify M given x^T only. Even worse, there is no guarantee that two models correctly describing x^T can yield the same values of y^T. For instance, in the morphological domain (Example 1), one can build degenerate functions ϕ such that $\phi(r) = x^T$ for all r. All such functions successfully describe x^T and yield all possible values for y^T.

The MML principle presented in Equation 5 is meant to be used in a single domain, in particular the source domain. We propose to use it as well to estimate the target model M^T. However, in the context of an analogy, some additional information is provided by the observation of the source domain, where a model M^S can be evaluated from observations $(x^S, y^S) \in \mathcal{D}^S$. The target model M^T can then be described with regards to the source model M^S.

The limitation of such a relative description of models is that the source model may not be relevant. In the following, we propose to quantify this relevance with two measures of model reusability,

3.2. Reusability of a Source Model

We measure the reusability of a source model to its ability to compress the information about the target domain. We identify two main possible definitions, that we call *weak* and *strong* reusability, and will show that a strongly reusable model is necessarily weakly reusable.

The main criterion for *weak reusability* is that knowing the source model M^S makes the target inference better, in the sense that it compresses more the description of the target case $(x^T, y^T) \in \mathcal{D}^T$.

Definition 1 (Weak reusability). *Let $\eta > 0$. A model $M^S \in \mathbb{M}_{\mathcal{R}^S, \mathcal{D}^S}$ is called* weakly η-reusable *for case (x^T, y^T) in target model space $\mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T}$ if:*

$$\begin{aligned} &\min_{M \in \mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T}} \{K(M) + K(x^T, y^T | M)\} \\ &\qquad \geq \eta + \min_{M \in \mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T}} \{K(M | M^S) + K(x^T, y^T | M)\} \end{aligned} \tag{6}$$

It is essential to keep in mind that this definition is relative to the choice of a model space $\mathbb{M}_{\mathcal{R}^T,\mathcal{D}^T}$ for the target domain. A source model M^S may not be reusable for a case (x^T, y^T) depending on the chosen target model space. In the following, we will abusively omit to mention the target model space, for readability purposes.

Note that the models M defined in the left-hand side and in the right-hand side of inequality (6) are not the same. On the left-hand-side, the model corresponds to the optimal target model describing (x^T, y^T) while, on the right-hand-side, it is the optimal model describing (x^T, y^T) when M^S is known.

This notion of reusability means that providing the source model M_S helps finding a new description of the problem shorter than the optimal description by η bits. In particular, in the case where $\mathbb{M}_{\mathcal{R}^S,\mathcal{D}^S} = \mathbb{M}_{\mathcal{R}^T,\mathcal{D}^T}$, the optimal model for (x^T, y^T) (i.e. the model that minimizes $K(M) + K(x^T, y^T|M)$) is trivially weakly reusable for (x^T, y^T) for all η. In other words, when the optimal model for the source domain is also optimal for the target domain, then it is obviously reusable for the target domain. The interesting case will be when the optimal source model is not optimal for the target domain.

We notice that the definition does not require $\mathcal{D}^S = \mathcal{D}^T$ and aims to quantify the reusability of a model even for a completely different task. This is possible since complexity only requires to have computable models: Indeed, the term $K(M|M^S)$ is defined as long as the models are computable. In practice, the issue of comparing objects of different nature is hidden within the choice of the reference p.r.p. function for complexity (see Section 2.2).

We propose an alternative definition of reusability, called *strong reusability.* It is based on the idea that M_S is reuseable if it helps compressing the optimal model of target case (x^T, y^T). Unlike previous definition, M_S is not directly involved in the description of (x^T, y^T) though.

Definition 2 (Strong reusability)**.** *Let* $\eta > 0$*. A model* $M^S \in \mathbb{M}_{\mathcal{R}^S,\mathcal{D}^S}$ *is called* strongly η-reusable *for case* (x^T, y^T) *in target model space* $\mathbb{M}_{\mathcal{R}^T,\mathcal{D}^T}$ *if:*

$$M \in \underset{M \in \mathbb{M}_{\mathcal{R}^T,\mathcal{D}^T}}{\mathit{arg}\min} \{K(M) + K(x^T, y^T|M)\} \\ \implies K(M) \geq K(M|M^S) + \eta \tag{7}$$

This definition also relies on the choice of a target model space. As for weak reusability, we will abusively omit the model space in the following notations.

Strong reusability is an extremely strong property of a source model. Indeed, it would be a natural property that, in case $\mathbb{M}_{\mathcal{R}^S,\mathcal{D}^S} = \mathbb{M}_{\mathcal{R}^T,\mathcal{D}^T}$, any model minimizing $K(M) + K(x^T, y^T|M)$ is reusable to (x^T, y^T). This is not the case for strong reusability: indeed, the definition implies that *any* model minimizing $K(M) + K(x^T|M)$ can be compressed given M^S, not only M^S itself. Note that this could be alleviated by weakening Definition 2 and requiring only the existence of a model M minimizing $K(M) + K(x^T, y^T|M)$ and such that $K(M) \geq K(M|M^S) + \eta$.

3.3. Properties of Reusability

We now present basic properties of these two notions of reusability. All the presented properties follow directly from the definitions.

The first property applies to both weakly and strongly reusable models: it states that the threshold η is not unique. The proof of this proposition is trivial and omitted.

Proposition 1. *Let $M_S \in \mathbb{M}_{\mathcal{R}^S,\mathcal{D}^S}$ and $(x^T, y^T) \in \mathcal{D}^T$. If M^S is η-reusable for (x^T, y^T), then it is also η'-reusable for (x^T, y^T) for all $\eta' \leq \eta$. In the following, we will call* degree of reusability *of M^S to (x^T, y^T) the quantity:*

$$\rho(M^S, (x^T, y^T)) = \max\left\{\eta\ ;\ M^S \text{ is } \eta\text{-reusable for } (x^T, y^T)\right\} \tag{8}$$

However, we insist on the fact that the degree of reusability is relative to a chosen target model space. It also depends on which of weak or strong reusability is considered. This dependency would not exist if these two notions of resuability were equivalent, which we will see is not the case. We will use the notation ρ_w and ρ_s to specify between the weak and strong cases.

We now show that strong reusability implies weak reusability:

Proposition 2. *Let $M^S \in \mathbb{M}_{\mathcal{R}^S,\mathcal{D}^S}$ a source problem, a target case $(x^T, y^T) \in \mathcal{D}^T$ and $\eta > 0$. If M^S is strongly η-reusable for (x^T, y^T), then M^S is also weakly η-reusable for (x^T, y^T).*

Proof. We call $M^* \in \mathbb{M}_{\mathcal{R}^T,\mathcal{D}^T}$ an optimal model for (x^T, y^T): $M^* \in \arg\min_M\{K(M) + K(x^T, y^T|M)\}$. Under the assumptions of the proposition, it follows that:

$$\begin{aligned}
&\min_M\{K(M) + K(x^T, y^T|M)\} \\
&\qquad = K(M^*) + K(x^T, y^T|M^*) \\
&\qquad \geq K(M^*|M^S) + \eta + K(x^T, y^T|M^*) \\
&\qquad \geq \min_M\{K(M|M^S) + K(x^T, y^T|M) + \eta\}
\end{aligned}$$

which proves the proposition. □

However, the converse is not true: weak reusability does not imply strong reusability. This is the consequence of the fact that the models implied in Equation (6) are not the same on the right hand side and on the left-hand side. We illustrate this with a simple example.

Example 3. *We consider a target model space made up of two distinct models: $\mathbb{M}_{\mathcal{R}^T,\mathcal{D}^T} = \{M_1, M_2\}$. We assume the following properties:*

- *M_1 and M_2 describe equally well case (x^T, y^T), in the sense that $K(x^T, y^T|M_1) = K(x^T, y^T|M_2)$*
- *Model M_1 is more complex than model M_2: $K(M_1) > K(M_2)$*
- *Model M_1 is easily described by source model M^S: for simplicity, we can take $K(M_1|M^S) = 0$*
- *Model M_2 is not well described by source model M^S: for simplicity, we can take $K(M_2|M^S) = K(M_2)$*

With these assumptions, it can be easily verified that M^S is weakly η-reusable for (x^T, y^T), with $\eta \leq K(M_2)$. However, M^S is not strongly η-reusable for (x^T, y^T), since M_2 minimizes $K(M) + K(x^T, y^T|M)$ but $K(M_2) < K(M_2) + \eta = K(M_2|M^S) + \eta$.

Putting together the results of Propositions 1 and 2, as well as the counter-example of Example 3, we can establish the following result:

Corollary 1. *For $M_S \in \mathbb{M}_{\mathcal{R}^S,\mathcal{D}^S}$ and $(x^T, y^T) \in \mathcal{D}^T$:*

$$\rho_s(M^S, (x^T, y^T)) \leq \rho_w(M^S, (x^T, y^T)) \tag{9}$$

3.4. Transferable Cases

The notion of reusability we proposed in Definitions 1 and 2 are not directly applicable to measure transferability from a source observation to a target problem. The property of transferability measures the ability to transfer knowledge from a source case $(x_S, y_S) \in \mathcal{D}^S$ to apply it to the target case $(x^T, y^T) \in \mathcal{D}^T$. It can be seen as an extension of reusability where the source model M^S is determined based on the source observation.

Definition 3 (Transferability)**.** *Let $(x^S, y^S) \in \mathcal{D}^S$ be a source case. The case (x^S, y^S) is said to be strongly (resp. weakly) η-transferable to the target case $(x^T, y^T) \in \mathcal{D}^T$ if the set of compatible models $\{M^S | K(M^S) + K(x^S, y^S|M_S) < K(x^S, y^S), M^S \in \mathbb{M}_{\mathcal{R}^S,\mathcal{D}^S}\}$ contains an element M^{S*} such that M^{S*} is strongly (resp. weakly) reusable for case (x^T, y^T).*

The proposed definition of transferability is very weak, since it only requires the existence of one source model that is reusable. However, it does not take into account the quality of this model in the source domain. For instance, it would acquire equal weight if the reusable source model is associated with complexity $K(M^S) + K(x^S, y^S|M^S)$ close to $K(x^S, y^S)$, as if the complexity is close to 0. Such situations are very likely to occur with probabilistic models, since many such models give positive probability to all observations.

In order to refine this notion, it would be important to define a coefficient of transferability, similar to the degree of reusability ρ defined in Proposition 1. We will denote such a coefficient $\tau((x^S, y^S), (x^T, y^T))$. We propose two possible definitions below. These definitions rely on the set of compatible models for (x^S, y^S) introduced in Definition 3, and denoted $\mathbb{M}(x^S, y^S)$. For simplicity, we will use the notation $\mathcal{C}^S$ (resp. $\mathcal{C}^T$) to designate the case (x^S, y^S) (resp. (x^T, y^T)).

A first definition is a direct application of Definition 3: it associates the transferability of a problem to the maximum reusability of the corresponding model:

$$\tau_{\max}\left(\mathcal{C}^S, \mathcal{C}^T\right) = \max_{M^S \in \mathbb{M}(\mathcal{C}^S)} \rho(M^S, \mathcal{C}^T) \tag{10}$$

with the convention that $\max \emptyset = 0$. This definition has the same weakness as the introduced concept of transferability: it does not take into account that the most reusable model might also be very weak to describe (x^S, y^S).

In order to alleviate this, our second definition proposes to average the score over the possible models. Therefore, we compute the posterior of the model for given (x^S, y^S) and compute the score as the expected value over M_S of the degree of reusability.

$$\tau_{\text{avg}}\left(\mathcal{C}^S, \mathcal{C}^T\right) = \sum_{M^S \in \mathbb{M}(\mathcal{C}^S)} p(M^S|x^S, y^S)\rho(M^S, \mathcal{C}^T) \tag{11}$$

In order to compute the posterior $p(M^S|x^S, y^S)$, we use algorithmic probability [10] and assume a uniform prior:

$$p(M^S|x^S, y^S) = 2^{-K(x^S, y^S|M^S) - K(M^S)} \tag{12}$$

This second definition provides a better idea of how transferable a source observation can be, since it takes into account all possible models describing it. However, it has two major weaknesses. On a theoretical level, it does not reflect the variance of the reusability degree over the compatible models: based on ρ_{avg} only, it is impossible to know whether all models, only very few but very compatible models, or a large number of less compatibles are reusable. On a practical level, the τ_{avg} score may not be tractable, and would require some approximations.

4. Illustration: Transferability of Morphological Transformations

In this section, we propose to apply the notions introduced in Section 3 to the case of morphological analogies. We will mostly build upon the domain introduced in Example 1.

4.1. Introduction to Morphological Analogies

Morphological analogies are analogies on words involving transformations of a morphological nature, for instance declension or conjugation. Unlike semantic analogies (e.g. “king is to queen as man is to woman"), morphological analogies are mostly of a symbolic nature: they involve the detection of transformations of one form of a word into another form. Typical example of such morphological analogies could be “work : worked :: call : called" in English (transformation from present to preterit, in English), or “voihin : vuossa :: soihin : suossa" (transformation from illative plural to inessive singular in Finnish).

Various works have been proposed to solve such analogies, mostly based on the principles of proportional analogy [7]. Other approaches rely on an algebraic consequence of these principles [16], on deep learning approaches [17] or even on complexity minimization [18].

4.2. A Simplified Model for Morphology

The model proposed in Example 1 is relevant for a formal treatment of morphological analogies. However, following Murena et al. [18], we propose a simplification where the space of partial recursive functions is restricted to a simpler subset. This subset is defined by a simple descriptive language based on the concatenation of various strings. It allows to define functions of the form $\phi = (\phi_1, \phi_2)$ with, for $i \in \{1, 2\}$:

$$\phi_i(r_1, \ldots, r_n) = w_0^i + \sum_{k=1}^{K_i} r_{\sigma_i(k)} + w_k^i \tag{13}$$

where the + operation stands for string concatenation, $r_1, \dots, r_n \in \mathcal{A}^*$ and $w_0^i, \dots, w_K^i \in \mathcal{A}^*$ are words on the alphabet $\mathcal{A}$, and $\sigma_i : \{0, \dots, K_i\} \to \{0, \dots, n\}$. The vector $r = (r_1, \dots, r_n)$ corresponds to the representation, and the models are defined such as in Equation (1).

We notice that the proposed restriction accounts for various types of morphological transformations, such as prefixation, suffixation, change or prefix and/or suffix, but also duplication. However, the language is not Turing complete, and for instance does not cover any conditional statement.

In morphological analogies, the source and target domains are the same.

4.3. Computing Complexities

In order to compute the reusability and transferability, we must define three expressions of complexity: complexity of a model $K(M)$, complexity of a case given a model $K(x, y|M)$ and complexity of a target model given a source model $K(M^T|M^S)$.

A model is entirely defined by the function ϕ. A binary coding of such functions is proposed in [18]. The complexity of the model then corresponds to the length (in bits) of the binary code of the function.

Given a model M, the case (x, y) is coded by providing a correct representation $(r_1, \dots, r_n)$. In case no representation generates (x, y) with model M, the two words are hard-coded. In terms of binary representation, we propose the following coding. The string starts with a bit encoding whether the following bits code for the representation vector or for the two words. After this bit, the words are coded letter by letter, with specific delimitors to mark the end of each string. The complexity $K(x, y|M)$ is the length of this code.

For the model transfer, we propose an elementary description. More sophisticated versions have to be discussed in future works. We propose to reuse the source model either by reusing it directly (without modification), or by redefining completely. Such as previously, this choice is indicated by an initial bit. The model complexity is then given by:

$$K(M^T|M^S) = \begin{cases} 1 & \text{if } M^T = M^S \\ 1 + K(M^T) & \text{otherwise} \end{cases} \tag{14}$$

4.4. Examples of Reusability

Suffixation. We consider the source model M^S associated to transformation $\phi(r_1) = (r_1, r_1 + \text{“}s\text{”})$ which consists in suffixing an “s" at the end of a word. We can verify that M^S is η-reusable for the target case (“film", “films"). In that case, it can be verified that M^S minimizes quantity $K(M) + K(x^T, y^T|M)$. Consequently, M^S is weakly η-reusable for (“film", “films") with $\eta \leq K(M^S) - 1$. This shows that $\rho_w(M^S, (\text{“film", “films"})) = K(M^S) - 1$. It can also be verified that M^S is the *only* model minimizing $K(M) + K(x^T, y^T|M)$. Therefore, we also have that M^S is strongly η-reusable for (“film", “films") with $\eta \leq K(M^S) - 1$. In this case, we have $\rho_s(M^S, (\text{“film", “films"})) = \rho_w(M^S, (\text{“film", “films"}))$.

However, the model is not reusable for (“mouse", “mice") for instance, since the minimal transformation for this case is $\phi'(r) = (r + \text{“ouse"}, r + \text{“}ice\text{”})$. We then have $\rho_s(M^S, (\text{“mouse", “mice"})) = \rho_w(M^S, (\text{“mouse", “mice"}))$.

Duplication. We consider the source model M^S associated to transformation $\phi(r_1) = (r_1 + \text{“} - \text{”} + r_1, r_1)$. This transformation is the reverse of a duplication (plural form in Indonesian). As for previous example, one can easily check that M^S is not reusable for the case (“orang", “orang-orang"). This was expected, with the choice of the transfer representation $K(M^T|M^S)$ which forces toward reusing the source model or ignoring it completely. It would not have been the same with other choices though. In particular, if it allowed for model transformation of the form $(\phi_1, \phi_2) \mapsto (\phi_2, \phi_1)$.

5. Toward Transferability in Domain Adaptation

Domain adaptation is a machine learning task where a hypothesis on a source domain has to be transferred to a target domain where data are not equally distributed [19]. We conclude this paper with a quick investigation of how domain adaptation can fit to our proposed framework.

5.1. Related Works: Task-Relatedness

The question of transferability of one solved source problem to a target problem has played a predominant role in the theoretical understanding of domain adaptation. Ben-David et al. (2010) [20] propose a PAC bound for transfer between domains in a binary classification setting, which relies on the use of a measure of a specific *domain divergence*, called $\mathcal{H}$-divergence. It is noticeable that the introduced measure depends on the hypothesis class $\mathcal{H}$, i.e. on the model space. This proposed notion has then been refined, to account for a variety of loss functions [21] or to adapt to the PAC-Bayesian setting [22]. All these measures follow a similar idea of comparing the distribution over the input spaces, but ignore the labels. Closer to our proposal, Zhang et al. (2012) [23] propose to additionally take into account the label distribution $p(y)$ in the discrepancy measure. Independently from these studies, Mahmud (2009) [24] also proposed to quantify the task-relatedness using Kolmogorov complexity and Algorithmic Information Theory.

5.2. Open Question: Complexities for Probabilistic Models

We describe the domain adaptation task in the context of the probabilistic model space defined in Example 2, in which a model is associated to a probability distribution. Such as for the morphological domain (Section 4.3), we need to define the complexities $K(M)$, $K(x, y|M)$ and $K(M^T|M^S)$.

The term $K(x, y|M)$ is a standard quantity considered by Algorithmic Information Theory. It is commonly computed using the *Shannon-Fano coding*. Using this allows to define the complexity of an object (x, y) knowing a probability distribution p as $K(x, y|p) = -\log p(x, y)$. This corresponds to the natural choice when computing $K(x, y|M)$ in our probabilistic domain.

Defining $K(M)$ and $K(M^T|M^S)$ is more difficult thought and goes beyond the scope of this paper. Traditionally, the complexity of a probability distribution is assimilated to the probability of its density function, and specific computations are proposed to estimate these. We refer the interest the interested reader to [24] for instance.

5.3. Discussion

Extending our framework to domain adaptation is both natural and complex. Indeed, the problem formulation in terms of models allow for a direct characterization of domain adaptation, where a domain is characterized by an unlabeled dataset (the problem) and a vector of predictions (the solution). However, the computation of the complexities may not be as simple as for the symbolic domain. Future works will have to bridge this gap.

6. Conclusion

In this paper, we introduced a novel understanding of analogical transfer, by focusing on whether the information contained in the source case were transferable to the target case. We proposed a formalism based on a notion of *models*, which we defined in a way that is consistent with both symbolic analogies and numerical machine learning. Even though this preliminary works cover only the question of measuring the transferability from a source case (x^S, y^S) to a target case (x^T, y^T), it is a first step toward the key question of predicting whether knowledge of (x^S, y^S) could be reused to find a solution y to problem x^T. We think the proposed framework could provide a common basis to develop a general understanding of transfer, going beyond the current statistical theories [20] for instance.

Acknowledgments

The author wishes to thank the anonymous reviewers for their insightful comments and suggestions. Some of the presented ideas have been inspired by discussions with Antoine Cornuéjols, Jean-Louis Dessalles, Marie Al-Ghossein and Miguel Couceiro. This work was supported by the Academy of Finland Flagship programme: Finnish Center for Artificial Intelligence, FCAI.

References

[1] D. Gentner, C. Hoyos, Analogy and Abstraction, Top. Cogn. Sci. 9 (2017) 672–693.

[2] L. Miclet, H. Prade, Handling analogical proportions in classical logic and fuzzy logics settings, in: G. C. C. Sossai (Ed.), Proc. 10th Europ. Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2009), Springer, Berlin/Heidelberg, 2009, pp. 638–650.

[3] N. Barbot, L. Miclet, H. Prade, Analogy between concepts, Artif. Intell. 275 (2019) 487–539.

[4] D. Hofstadter, M. Mitchell, Fluid Concepts and Creative Analogies, Basic Books, Inc., New York, NY, USA, 1995, pp. 205–267.

[5] 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, 2013, pp. 746–751.

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

[7] Y. Lepage, Analogy and Formal Languages, Electron. Notes Theor. Comput. Sci. 53 (2001) 180–191.

[8] P. A. Murena, J.-L. Dessalles, A. Cornuéjols, A complexity based approach for solving Hofstadter's analogies, in: CAW@ ICCBR-2017 Computational Analogy Workshop, at International Conference on Case Based Reasoning, 2017.

[9] P.-A. Murena, M. Al-Ghossein, Inferring Case-Based Reasoners' Knowledge to Enhance Interactivity, in: International Conference on Case-Based Reasoning, Springer, 2021, pp. 171–185.

[10] M. Li, P. Vitányi, et al., An introduction to Kolmogorov complexity and its applications, volume 3, Springer, 2008.

[11] C. Antić, Analogical Proportions, 2020. URL: https://arxiv.org/abs/2006.02854.

[12] S. Shalev-Shwartz, S. Ben-David, Understanding machine learning: From theory to algorithms, Cambridge university press, 2014.

[13] C. S. Wallace, D. M. Boulton, An information measure for classification, The Computer Journal 11 (1968) 185–194.

[14] J. Rissanen, Modeling by shortest data description, Automatica 14 (1978) 465–471.

[15] P. D. Grünwald, The minimum description length principle, MIT press, 2007.

[16] P. Langlais, F. Yvon, P. Zweigenbaum, Improvements in analogical learning: application to translating multi-terms of the medical domain, in: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), 2009, pp. 487–495.

[17] 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.

[18] P.-A. Murena, M. Al-Ghossein, J.-L. Dessalles, A. Cornuéjols, Solving Analogies on Words based on Minimal Complexity Transformation., in: IJCAI, 2020, pp. 1848–1854.

[19] A. Farahani, S. Voghoei, K. Rasheed, H. R. Arabnia, A brief review of domain adaptation, Advances in Data Science and Information Engineering (2021) 877–894.

[20] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. W. Vaughan, A theory of learning from different domains, Machine learning 79 (2010) 151–175.

[21] Y. Mansour, M. Mohri, A. Rostamizadeh, Domain adaptation: Learning bounds and algorithms, arXiv preprint arXiv:0902.3430 (2009).

[22] P. Germain, A. Habrard, F. Laviolette, E. Morvant, A PAC-Bayesian approach for domain adaptation with specialization to linear classifiers, in: International conference on machine learning, PMLR, 2013, pp. 738–746.

[23] C. Zhang, L. Zhang, J. Ye, Generalization bounds for domain adaptation, Advances in neural information processing systems 25 (2012).

[24] M. H. Mahmud, On universal transfer learning, Theoretical Computer Science 410 (2009) 1826–1846.