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. $a : b :: c : d \rightarrow c : d :: a : b$ (*symmetry*);
3. $a : b :: c : d \rightarrow a : c :: b : d$ (*central permutation*).

These postulates have straightforward consequences like:

- $a : a :: b : b$ (*identity*);
- $a : b :: c : d \rightarrow b : a :: d : c$ (*internal reversal*);
- $a : b :: c : d \rightarrow d : b :: c : a$ (*extreme permutation*);
- $a : b :: c : d \rightarrow d : c :: b : a$ (*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":

$a : b :: c : d \rightarrow b : a :: d : c$ (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. $a : b :: c : d \rightarrow c : d :: a : b$ (*symmetry*);
3. $a : b :: c : d \rightarrow b : a :: d : c$ (*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:

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

with $+$ 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

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

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.