

Embedding-To-Embedding Method Based on Autoencoder for Solving Sentence Analogies

Weihao Mao^{1,*}, Yves Lepage¹

¹Graduate School of Information, Production and Systems, Waseda University

Abstract

We propose a method for solving sentence analogies using an embedding-to-embedding method. The method involves the pretraining of an autoencoder with a denoising decoder that generates sentence embeddings and reconstructs sentences. To generate solutions to analogical equations in the sentence embedding space, we introduce a network architecture that learns analogy properties from the dataset instead of relying on predefined formulas. The embeddings of the solutions are then decoded back into sentences using the decoder of the pretrained autoencoder. We conduct experiments on a set of semantico-formal analogies and purely-formal analogies datasets in English, French, and German. The results show that our method achieves state-of-the-art performance in most cases and to some extent provides evidence of the limitations of the 3CosAdd formula in handling longer sentences.

Keywords

Sentence analogy, Sentence embedding, Autoencoder

1. Introduction

Analogical reasoning is a central process of human cognition that helps us explain and describe abstract concepts by comparing representations of things and retrieving potential similarities in memory. It shares similarities with case-based reasoning (CBR), as both involve inferring new facts or answers based on the nature of similar existing facts. However, case-based reasoning places more emphasis on the implementation details and logic of the reasoning process in computer systems. Therefore, research advancements in analogical reasoning are also beneficial for case-based reasoning. Analogical reasoning can be explained as being based on analogy, a relationship between four objects A , B , C , and D . It can be simply described as " A is to B as C is to D ," written as $A : B :: C : D$. Two of the main concepts are *ratio* and *conformity*. The ratio can be interpreted as the relationship between A and B or between C and D , while conformity ensures that the relationship (ratio) between the two members of the analogy is consistent.

In the two simple examples below, the ratio in the first example is only a formal substitution of some words in the sentence. For instance, we can clearly observe that *us* is replaced by *me*. This type of analogy is called a formal analogy. The ratio of the second example contains

ICCBR ATA'23: Workshop on Analogies: From Theory to Applications – AR & CBR Tools for Metric and Representation Learning at ICCBR2023, July 17 – 20, 2023, Aberdeen, Scotland

*Corresponding author.

✉ mao_weihao@asagi.waseda.jp (W. Mao); yves.lepage@waseda.jp (Y. Lepage)

🆔 0009-0002-3103-8012 (W. Mao); 0000-0002-3059-4271 (Y. Lepage)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

please tell us about it. : *please tell me about it.* :: *what do you expect us to do?* : *what do you expect me to do?*
he never saw his brother again. : *he never saw his sister again.* :: *he never saw his father again.* : *he never saw his mother again.*

semantic changes, such as the *brother* corresponding to the *sister* and the *father* corresponding to the *mother*. The ratio in this example contains some gender-related information. We refer to such examples as semantic analogies.

The above two important concepts indicate that we can deduce the fourth term based on any three terms of the quadruplet. This property has led to the gradual application of analogy to some natural language processing related tasks, such as natural language inference, question answering, and machine translation, especially EBMT (Example-based machine translation).

The application of analogies in natural language processing mainly involves two tasks that need to be addressed. The first one is *analogy detection*, i.e., determining whether a quadruple A , B , C , and D constitute an analogy. Since the concept of analogy still lacks a standard definition, we mainly refer to the analogy property proposed by [1], which is taken as an analogy if it satisfies the two properties of *symmetry of conformity* and *exchange of means*. Because we can reason out eight equivalent forms of an analogy based on these two properties. It is worth mentioning that such assumption provides a relatively strict definition for analogies, especially for sentence analogies. [2] introduces *internal reversal* as a substitution for the *exchange of means* mentioned above, allowing for more quadruples to meet the definition of this analogy at the sentence level.

The second primary task is *analogy solving*, the process of giving A , B , and C in a quadratic group to obtain D . That means we need to find the solution to the analogical equation:

$$A : B :: C : x \\ \Rightarrow x = ?$$

Currently, in recent years, methods mainly rely on vector representations of sentences in embedding space. The approach involves using the *parallelogram rule* (if $e_B - e_A = e_D - e_C$, then $e_D = e_B - e_A + e_C$) to find four vectors that satisfy the analogy property and simultaneously find the solution of the analogical equation in the embedding space.

After obtaining the embeddings of the solutions in the embedding space, a commonly used approach is to employ *retrieval-based* methods. These methods involve providing a set of candidate sentences and retrieving the most similar sentence to the target based on metrics like cosine similarity. One example of such a method is the *3CosAdd* method [3, 4]. These methods typically require the embedding space to exhibit good linearity properties and rely on specific formulas. They are unable to learn the analogy properties from the dataset itself. However [5] trains a decoder to map the embeddings of the solutions of analogical equations back to their corresponding sentences, which allows the model to generate results beyond the limitations of specific candidate sentences. We refer to these methods as *generation-based* methods. In generation-based methods, the model learns to generate sentences based on the given analogical equations, providing more flexibility in producing diverse and contextually appropriate outputs.

Inspired by the work of [5], we design a generative method based on an autoencoder to address sentence analogies. More precisely, the main contributions of this paper are as follows:

- i We have designed a more stable autoencoder architecture to reconstruct the solutions of analogical equations from the embedding space back into sentences.
- ii We propose a novel model that does not rely on predefined formulas to solve analogical equations in the sentence embedding space. The entire network architecture is more flexible and applicable to all encoder-decoder structures.
- iii We have achieved promising results in the generation-based approach and, to some extent, demonstrated that the effectiveness of the 3CosAdd formula decreases for longer sentences.

In the remaining sections of this paper, we first introduce the related work in solving analogies, particularly sentence analogies, in Section 2. In Section 3, we describe the main approach we adopt, namely the embedding-to-embedding method. In Section 4, we present the experiments and results. In Section 5, we provide an overview of the contributions of this paper and propose further directions for future research.

2. Related work

In this paper, we primarily focus on solving sentence analogies, which involve deriving an unknown sentence D given known sentence analogies A , B , and C . However, we can still draw inspiration from recent word analogy tasks. As mentioned in Section 1, some retrieval-based methods like *3CosAdd* rely on predefined formulas and expected properties of the embedding space. Their goal is not to learn the properties of analogies from existing actual data so as to solve analogy. [6] used a simple network architecture called *ANNr* that consists of only linear fully connected layers to learn the embeddings of words A , B , and C to D in the embedding space, rather than relying on predefined formulas. The model has achieved state-of-the-art performance on word analogy tasks in 11 different languages. This demonstrates that even without relying on traditional formulas such as $e_D = e_B - e_A + e_C$, but instead learning relevant properties from the dataset, one can achieve good results.

Unlike word analogy tasks, sentence analogies are more diverse and complex in terms of vocabulary, syntax, and semantics, making them more challenging to solve. However, a sentence can still be seen as a whole composed of multiple words. [7] proposed a method that decomposes sentence analogies into multiple sets of word analogies based on the editing traces between sentences. The optimal solutions of multiple sets of word analogies are then concatenated to form the solution for the sentence analogy. Indeed, this work has also resulted in the creation of a sentence *semantico-formal analogy dataset*.

[5] proposed a *Vec2Seq* model to learn the mapping from sentence vectors to corresponding sentences, thus addressing the limitation of retrieval-based approaches that can only select the best sentence from candidate sentences. This led to the idea of a generation-based solution. They first employed a simple sum operation of *FastText* [8] word vectors in corresponding dimensions to represent the entire sentence vector. Then, they trained a decoder to reconstruct the sentence

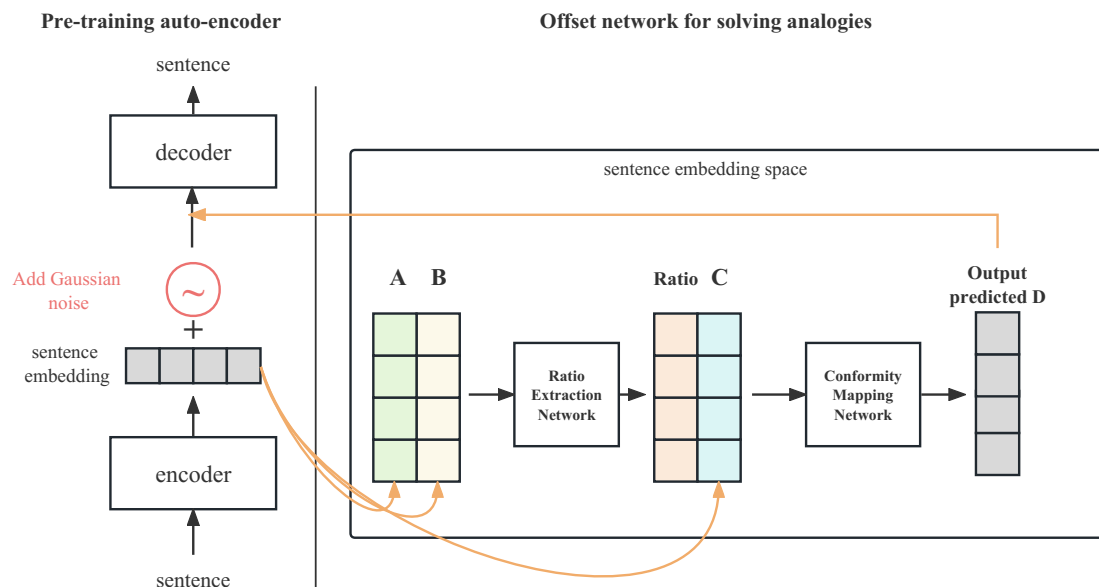


Figure 1: The diagram of the embedding-to-embedding method for solving sentence analogies

from the sentence vector. Additionally, they designed a simple linear fully connected network *FCN* to learn the mapping of analogical equation solutions in the embedding space. They tested different ways of combining vectors in *semantico-formal analogy dataset*, and the calculation formula $e_D = e_B - e_A + e_C$ in 3CosAdd ultimately achieved the best performance.

Inspired by work of [5], [9] proposed a *character-level autoencoder* to reconstruct words and address word analogy problems. This method achieved 99% accuracy on word reconstruction tasks in multiple languages and showed promising results in solving word analogy tasks.

3. Proposed approach

Similarly as in [5], we propose an internally denoising autoencoder architecture to achieve the generation of sentence vectors from word vector sequences and a more stable decoding process. Additionally, we introduce an offset network structure to learn the mapping from three known vectors to a solution of analogical equations in the sentence embedding space. As this approach operates in the sentence embedding space, it is referred to as an "embedding-to-embedding" method. The entire method architecture is illustrated in Figure 1.

3.1. Pre-training an auto-encoder

The method used in [5] for generating sentence vectors from word vector sequences involves simply adding up the corresponding dimensions of all word vectors to form the sentence vector. This method, starting from pre-trained word vectors, can produce decent decoding results even with a small amount of training data. Additionally, the simple addition of corresponding dimensions is quite effective for certain specific tasks. However, the sentence embeddings

generated by this simple summation method tend to lose sequential information and some semantic information. Structurally, this method is not conducive to sentence reconstruction. Therefore, taking inspiration from that method, we also start from pre-trained word vectors and retain its decoder part. However, we incorporate a bidirectional LSTM model as an encoder to process the word vector sequence and form an autoencoder structure. Subsequently, we adopt the method mentioned in [9] to obtain sentence embeddings, which involves concatenating the last hidden state and cell state of the encoder as the resulting sentence embedding.

Additionally, because the task of the decoder is to decode embeddings that satisfy the constraints of analogical equations, there may be slight deviations in the numerical values of the generated embeddings, whether produced by neural networks or predefined formulas, compared to the true reference embeddings. This can cause the decoder to struggle in correctly decoding these embeddings. Therefore, during the training process of the autoencoder, we introduce a certain proportion of Gaussian noise to the sentence embeddings generated by the encoder, aiming to train the decoder to produce accurate sentences. This approach enhances the decoder's robustness to small perturbations along the target embedding manifold, expands the range of manifolds the decoder can correctly decode, and mitigates overfitting to some extent.

3.2. Embedding-to-embedding method for solving analogies

After completing the pre-training of the autoencoder, we can obtain sentence embeddings using the well-trained encoder. Within the generated embedding space, we propose an Offset network structure to learn predicting embeddings that satisfy the constraints of analogical equations. This neural network is based on two important concepts of analogies: conformity and ratio, and it is divided into two parts: the ratio extraction network and the conformity mapping network.

The ratio extraction network, the first part of the Offset network, learns the ratio relationship in the analogy by taking the embeddings of sentences A and B as inputs.

The conformity mapping network, the second part, learns to map the ratio and the embedding of sentence C to obtain the embedding of sentence D .

These two parts of the network have a simple structure, consisting of only one layer of convolutional network and one fully connected layer. To some extent, this network structure achieves the offset of embedding C by ensuring the conformity of the ratio between two binary tuples in the analogy. Hence, we refer to it as the *Offset network*. Our expectation is that it can learn the properties of analogies from the dataset and solve analogies without relying on predefined formulas.

4. Experiments

4.1. Evaluation metrics

In our experiments, we want the generated sentences and the reference sentences to be as similar as possible. So we use BLEU [10] to evaluate the similarity of two sentences. BLEU scores are between 0 and 100. The higher the score, the more similar the two sentences are. We also use the Levenshtein distance to evaluate the degree of difference between two sentences.

Table 1

Statistics of the purely formal analogy datasets in the three languages.

data	Number of			
	analogies	sentences	words/sents.	character/sents.
English				
Taining	8,000	18,515	5.7±1.7	22.7±8.1
Validation	1,000	3,639	5.5±1.7	22.1±7.9
Testing	1,000	3,666	5.6±1.7	22.2±8.1
French				
Taining	8,000	14,803	7.0±2.7	29.7±12.3
Validation	1,000	3,482	7.0±2.9	30.1±12.8
Testing	1,000	3,478	7.0±3.0	30.1±13.4
German				
Taining	8,000	12,729	6.1±2.0	29.2±10.9
Validation	1,000	3,226	6.1±2.0	28.6±10.5
Testing	1,000	3,232	6.1±1.9	28.5±10.4

In addition, the accuracy rate is the ratio of the number of perfectly predicted sentences to the total number of reference sentences.

4.2. Data

For the pre-training of the auto-encoder, we randomly extracted 85,000 English, French, and German sentences from the Tatoeba¹ corpus. The average length of English sentences is 6.5, while for French and German, it is 8.7. We split into 80%, 10%, 10% for training, validation and testing. In order to evaluate our method for solving sentence analogies, we conducted tests on the *semantico-formal analogy dataset* proposed in [7], which contains 5,607 sentence analogies. Additionally, to further assess the performance of our model in solving sentence formal analogies, we utilized the Nlg package proposed in [11] to extract purely formal analogies from Tatoeba in the three languages. Statistics on the data are presented in Table 1.

4.3. Setups

For decoding sentence embeddings, we keep the decoder part of the autoencoder consistent with the decoder in [5]. After obtaining word vector sequences using pre-trained FastText word embeddings, we employ two approaches to obtain sentence embeddings, i.e., *simple summation*: adding the word vectors corresponding to each dimension together. *encoder of autoencoder*: using a bidirectional LSTM to obtain sentence embeddings. During training, we employed *cross-entropy* as the loss function and utilized the Adam optimizer with a learning rate to 0.001. In training the sentence embeddings decoder, we set the maximum iteration count to 1000 and used an early stopping mechanism, which means that training stops if there is no improvement after 15 iterations. However, for solving sentence analogies, we set the tolerance count for early stopping to 50. Additionally, when training the model for solving sentence analogies, we froze the parameters of the autoencoder, meaning that we did not fine-tune the embedding model.

¹<https://tatoeba.org/en/>

Table 2

Performance of the different models on three languages.

Input Vector composition method	Model size (Mb)	BLEU	Accuracy (%)	Levenshtein distance	
				in words	in cahrs
English					
simple summation	3.8	73.5±0.7	62.2	1.0	4.3
encoder of autoencoder	4.4	93.5±0.4	91.1	0.1	0.8
French					
simple summation	8.8	42.2±0.9	25.9	3.3	15.2
encoder of autoencoder	11.6	68.5±1.1	56.3	1.4	9.2
German					
simple summation	11.0	35.4±0.8	24.0	3.7	19.1
encoder of autoencoder	13.8	60.6±1.0	54.0	2.4	12.5

We conducted experiments to solve sentence analogies using different combinations of methods, i.e.,

- *sum-FCN*: Using the *FCN* network proposed in [5] in conjunction with the formula from *3CosAdd* to process embeddings as inputs to solve analogies and obtaining sentence embeddings by *simple summation*.
- *enc-FCN*: Obtaining sentence embeddings using an encoder and solving analogies using the *FCN* network in conjunction with the formula from *3CosAdd* to process embeddings as inputs.
- *enc-Offset*: Obtaining sentence embeddings using an encoder and solving analogies using our *Offset* network.
- *enc-ANNr*: Obtaining sentence embeddings using an encoder and solving analogies with the *ANNr* network used in [6].

During training, we employed *MSE* as the loss function.

4.4. Performance in decoding sentence embeddings

During the pre-training of the autoencoder, we set the ratio of Gaussian noise added to the sentence embeddings as 0.1. Additionally, the dimension of the sentence embeddings was set to 300. The results on the three languages are shown in Table 2. In terms of accuracy, using sentence embeddings generated by the encoder of the autoencoder outperforms the simple summation approach by nearly 30% in all three languages. For English sentences, which are shorter with a smaller vocabulary, the decoding accuracy reaches 91.1%. Additionally, the 0.1 Levenshtein distance indicates that, on average, less than one word is incorrect when decoding English sentences. As for French and German, which have longer sentence lengths and vocabulary sizes two to three times larger than that of English, the decoding performance decreases slightly, but the decoding accuracy using encoder-generated sentence embeddings still surpasses the simple summation approach by a considerable margin more than 30%.

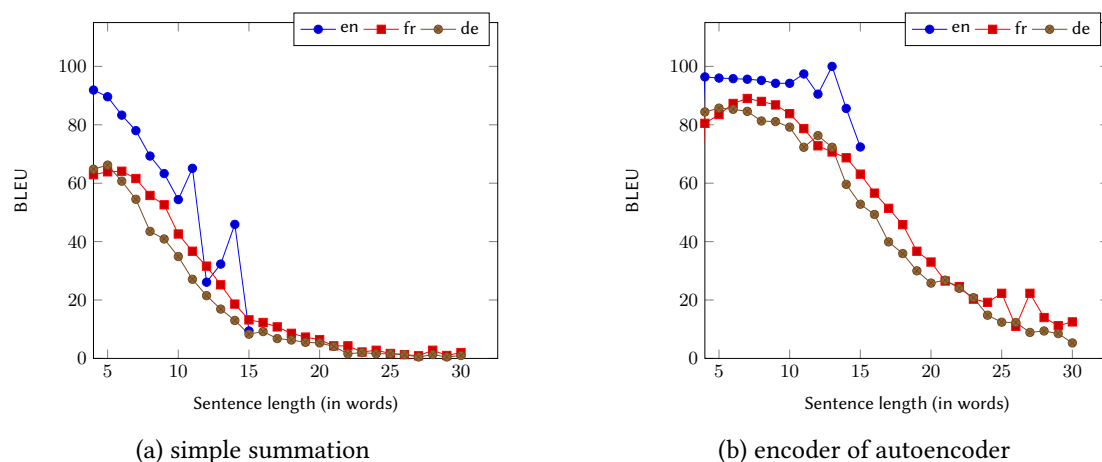


Figure 2: Performance of models on sentences with different lengths in three different languages

Additionally, we investigated the impact of sentence length on the decoding of sentence embeddings. As shown in Figure 2, we can observe that both methods experience a gradual decrease in decoding performance as sentence length increases. However, the approach of using encoder-derived sentence embeddings exhibits relatively more stability. Furthermore, due to the larger vocabulary in German, the decoding performance is relatively lower compared to the other two languages.

4.5. Performance in solving sentence analogies

4.5.1. Semantic-formal analogies

For the performance on the semantic-formal analogy set, we first evaluated the performance of our pre-trained autoencoder in decoding the embeddings of the fourth item sentence D in the analogy. Both accuracy and BLEU score reached 100, indicating that all the target reference sentences were perfectly reconstructed. Then, as described in Subsection 4.2, we tested the performance of different method combinations on this dataset. As shown in Table 3, from the perspective of obtaining sentence embeddings, using the encoder to obtain sentence embeddings performs similarly to the simple summing method, with only a 1-point difference in BLEU score. This is because the average length of English sentences in this dataset is relatively short, and both methods show similar performance in decoding sentence embeddings. However, from the perspective of solving analogies, the FCN network using the *3CosAdd* formula outperforms the *Offset* network and *ANNr*. This indirectly indicates that methods relying on predefined formulas are more effective than learning analogy properties from a dataset when the data size is limited, sentences are short, and analogies are relatively simple in form.

4.5.2. Purely formal analogies

For purely formal analogies, Table 4 presents the performance of different models in the three languages.

Table 3

Performance of the different models on semantico-formal analogy set.

Experiment name	BLEU	Accuracy (%)	Levenshtein distance	
			in words	in chars
sum-FCN	91.0±1.3	82.5	0.3	1.3
enc-FCN	92.0±1.3	84.6	0.2	1.0
enc-Offset	89.1±1.6	78.2	0.4	1.8
enc-ANNr	80.3±2.2	73.1	0.6	2.7

Considering the languages, although French and German have a larger vocabulary, the overall impact is mainly determined by the average sentence length. Since French has the longest average sentence length, followed by German, and English has the shortest, the performance of different models is generally lower in French compared to the other two languages. Especially, due to the extremely short average sentence length in English, when using the *FCN* network to solve analogies, the method of obtaining sentence embeddings using the encoder and the simple summing method show similar performance, with the simple summing method even outperforming it. In contrast, the performance trends of different models in French and German are roughly similar. First, the method of obtaining sentence embeddings using the encoder outperforms the simple summing method. Second, the *FCN* network performs better than the *Offset* network.

4.5.3. Performance on longer sentences

It is worth mentioning that French has a longer average sentence length, with the longest sentence reaching around 10 words. The performance of the *Offset* network is almost on par with the *FCN* network, with only a slight difference of around 1 in both BLEU score and accuracy. This suggests that when the average sentence length becomes longer, the *Offset* network, which learns analogy properties from the dataset, may perform well. Therefore, we further conducted tests on the French dataset by selecting sentences with a length of 10 or more, and the results are shown in Figure 3. We observe that when the sentence length exceeds 10, the *Offset* network performs better than *FCN*. We infer that when dealing with longer sentences, methods that learn analogy properties from the dataset are more reliable than using predefined formulas such as *3CosAdd*. This could be because the application of the *3CosAdd* formula in analogies of longer average sentence lengths requires the sentence embedding space to have more pronounced linear properties. On the other hand, learning from the dataset allows for lower expectations in the embedding space having linear properties, especially when there is a larger amount of data available.

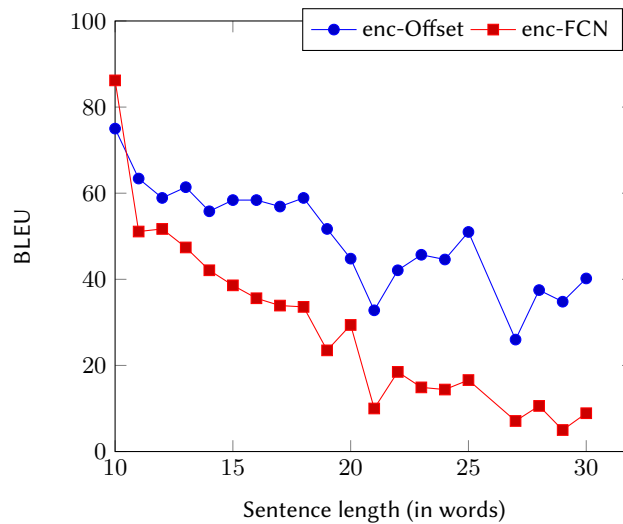
5. Conclusion

We proposed an auto-encoder architecture that internally removes noise to generate sentence embeddings and reconstructs sentences, achieving high accuracy in decoding sentence embeddings. Building upon this, we devised an embedding-to-embedding method and a model that

Table 4

Performance of the different models on formal analogy set in three languages.

Experiment name	BLEU	Accuracy (%)	Levenshtein distance	
			in words	in chars
English				
sum-FCN	91.0±1.8	90.8	0.3	1.0
enc-FCN	89.6±2.1	88.6	0.4	1.3
enc-Offset	80.6±2.2	76.1	0.7	2.4
French				
sum-FCN	64.3±2.6	46.2	1.7	7.5
enc-FCN	71.8±2.2	57.9	1.4	5.4
enc-Offset	70.6±2.2	56.1	1.5	6.2
German				
sum-FCN	73.6±2.3	62.3	0.9	3.8
enc-FCN	84.1±2.1	78.8	0.6	2.6
enc-Offset	77.0±2.3	69.0	0.8	3.6

**Figure 3:** Performance of models on sentences with different lengths in French

learns analogies from datasets in the sentence embedding space instead of relying on predefined formulas. Our experiments demonstrated that this approach performs better than a model relying on the *3CosAdd* formula, especially in cases where the sentence length is longer.

Our method for analogy solving is a generation-based approach. It is still limited by the drawback of LSTM decoders in handling long sentences. In the future, we need to explore more advanced encoder-decoder architectures that are better suited for decoding longer sentences, as well as generating more meaningful sentence embeddings specifically designed for analogies.

Acknowledgments

This research has been partially supported by a JSPS grant Kiban C n° 21K12038 entitled « Theoretically founded algorithms for the automatic production of test sets in NLP."

References

- [1] Y. Lepage, Languages of analogical strings, in: Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000), volume 1, Saarbrücken, 2000, pp. 488–494. URL: <https://aclanthology.org/C00-1071>.
- [2] S. Afantenos, S. Lim, H. Prade, G. Richard, Theoretical study and empirical investigation of sentence analogies, in: IJCAI-ECAI Workshop: Workshop on the Interactions between Analogical Reasoning and Machine Learning (IAMRL 2022)@ IJCAI-ECAI 2022, volume 3174, CEUR-WS. org, 2022, pp. 15–28.
- [3] 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, Association for Computational Linguistics, Atlanta, Georgia, 2013, pp. 746–751. URL: <https://aclanthology.org/N13-1090>.
- [4] O. Levy, Y. Goldberg, Dependency-based word embeddings, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 302–308. URL: <https://aclanthology.org/P14-2050>. doi:doi: 10.3115/v1/P14-2050.
- [5] 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:doi: 10.1109/ICACSIS51025.2020.9263191.
- [6] E. Marquer, S. Alsaïdi, A. Decker, P.-A. Murena, M. Couceiro, A deep learning approach to solving morphological analogies, in: M. T. Keane, N. Wiratunga (Eds.), Case-Based Reasoning Research and Development, Springer International Publishing, Cham, 2022, pp. 159–174.
- [7] Y. Lepage, Semantico-formal resolution of analogies between sentences, in: the 9th Language and Technology Conference (LTC 2019), 2019, p. 57–61.
- [8] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146. URL: <https://aclanthology.org/Q17-1010>. doi:doi: 10.1162/tacl_a_00051.
- [9] K. Chan, S. P. Kaszefski-Yaschuk, C. Saran, E. Marquer, M. Couceiro, Solving Morphological Analogies Through Generation, in: IJCAI-ECAI Workshop on the Interactions between Analogical Reasoning and Machine Learning (IARML@IJCAI-ECAI 2022), volume 3174 of *Proceedings of the IJCAI-ECAI Workshop on the Interactions between Analogical Reasoning and Machine Learning (IARML@IJCAI-ECAI 2022)*, Miguel Couceiro and Pierre-Alexandre Murena, Vienna, Austria, 2022, pp. 29–39. URL: <https://hal.inria.fr/hal-03674913>.
- [10] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association

- for Computational Linguistics, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 311–318. URL: <https://aclanthology.org/P02-1040>. doi:doi:10.3115/1073083.1073135.
- [11] R. Fam, Y. Lepage, Tools for the production of analogical grids and a resource of n-gram analogical grids in 11 languages, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018, pp. 1060–1066. URL: <https://aclanthology.org/L18-1171>.