HAL open science

Proceedings of the 2nd Workshop Analogies: from Theory to Applications (ATA@ICCBR 2023)

Fadi Badra, Miguel Couceiro, Esteban Marquer, Pierre Monnin

To cite this version:

Fadi Badra, Miguel Couceiro, Esteban Marquer, Pierre Monnin. Proceedings of the 2nd Workshop Analogies: from Theory to Applications (ATA@ICCBR 2023). 31st International Conference on Case-Based Reasoning (ICCBR 2023), CEUR Workshop Proceedings, 3438, pp.2-53, 2023, ICCBR 2023 Workshop Proceedings. hal-04392016

HAL Id: hal-04392016

https://inria.hal.science/hal-04392016v1

Submitted on 12 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Analogies: from Theory to Applications

AR & CBR Tools for Metric and Representation Learning

Organizers:

Fadi Badra (Université Sorbonne Paris Nord, France)
Miguel Couceiro (Université de Lorraine, France)
Esteban Marquer (Université de Lorraine, France)
Pierre Monnin (Orange, France)

Program Committee:

Adrien Coulet	Jean Lieber	Henri Prade
Mehdi Kaytoue	Marie-Jeanne Lesot	Anes Bendimerade
Youcef Remil	Mathieu d'Aquin	Christophe Cerisara
Claire Gardent	Steven Schockaert	Yves Lepage
Myriam Bounhas	Sebastien Destercke	

The purpose of this workshop is to explore both foundational and application aspects of analogical reasoning by bringing together AI researchers of various fields such as case-based reasoning, machine learning, cognitive psychology, knowledge representation, discovery, and reasoning, as well as industrial practitioners with real-world data and applications. Computational analogy and case-based reasoning (CBR) are closely related research areas. Analogy research often focuses on modeling human cognitive processes and developing computational theories of analogical reasoning, whereas CBR tends to focus more on the conception and knowledge engineering issues that need to be dealt with when implementing analogical reasoning in a computer system. These two focuses are very complementary. As the theme of this year's ICCBR is the place of CBR among modern AI techniques, we are particularly interested in how new computational theories of AR research can help CBR revisit its foundations and play its role in "modern AI". In particular, it aims to address the following challenges: how to represent and maintain cases, how to take into account domain knowledge, how to represent and learn similarity metrics for specific tasks, how to represent and learn adaptation knowledge, and how to derive useful explanations. This asks for a thorough investigation of analogical reasoning in predicting complex solutions, in deriving explanations as well as of the role of analogies in representation learning, in the relation between CBR and attention mechanisms, in the development of case-based prediction algorithms beyond k-Nearest Neighbors, and in novel methods for similarity measure learning.

Resolution of Analogies Between Strings in the Case of Multiple Solutions

Xulin Deng, Yves Lepage

Waseda University, Japan

Abstract

The verification and resolution of formal analogies between strings focuses on the character sequences, disregarding the underlying semantics of the sequences. Our approach to these two tasks employs an algorithm based on edit distance. A previous version was limited in that it provided only a single solution for an analogy equation, even when multiple valid solutions existed. We enhance the algorithm to generate all possible solutions. The previous algorithm traversed edit distance matrices only once. Consequently, it could only yield one solution for an analogy puzzle, even in cases of multiple solutions were viable. In order to deliver all possible solutions for analogies, we introduce a recursive approach. By recursively exploring all traces in the edit distance matrices, our newer version is capable of generating and outputting all feasible solutions.

Keywords

Analogy, Multiple solutions, Data Generation, Strings

1. Background

In this paper, we deal with formal analogies between strings, i.e., sequences of characters, like the ones described in [1]. We do not address analogy learning, nor semantic analogies [2]. An analogy is a relation between four terms; it is noted by $A : B :: C : D$, commonly read as "A is to B as C is to D". As said above, in this paper, the terms will be strings.

Analogy between strings can be applied in the field of natural language processing for tasks like transliteration [3, 4], morphology [5, 6, 7] or even machine translation [8, 9]. Methods to solve analogies are basic functions in such previous work. Several approaches have been proposed [4, 10, 11].

In the previous work by [10], which serves as the foundation of this paper, the algorithm demonstrated a high level of accuracy in providing precise answers for the majority of cases, with the exception of instances involving reduplication and permutation because this algorithm lacked the capability to address these particular linguistic phenomena [12]. But, in addition, in scenarios where analogy puzzles have multiple potential solutions, the algorithm was limited to generating a single answer. Consequently, this limitation introduced inaccuracy was a handicap for comparison with other proposals.

In the work conducted by [11], a complexity-based algorithm for solving analogies is introduced. The authors provide a comprehensive table presenting the proportion of correct solutions

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

origamisama@akane.waseda.jp (X. Deng); yves.lepage@waseda.jp (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)

CEUR-WS.org/Vol-3438/paper_01.pdf

Language	Number of analogies	Method		
		Complexity [11]	Distance [10]	Shuffle [7]
Arabic	165,113	87.18%	93.33%	81.91%
Finnish	313,011	93.69%	92.76%	78.75%
Georgian	3,066,273	99.35%	97.54%	88.42%
German	730,427	98.84%	96.21%	95.42%
Hungarian	2,912,310	95.71%	92.61%	86.02%
Maltese	28,365	96.38%	84.72%	91.84%
Navajo	321,473	81.21%	86.87%	78.95%
Russian	552,423	96.41%	97.26%	95.46%
Spanish	845,996	96.73%	96.13%	94.42%
Turkish	245,721	89.45%	69.97%	70.06%
Total	9,181,112	96.41%	94.34%	87.93%

Table 1
Table copied from [11] showing the accuracies of three approaches to solving morphological analogies in various languages.

for a dataset of analogy equations, compiled by [13] from the Sigmorphon Analogy Dataset, and recognized as the most extensive dataset available for this purpose. Table 1 reproduces this table. It gives the accuracy per language.

The Shuffle algorithm proposed by [4] is another algorithm employed for solving analogies. On the other hand, the Distance algorithm being referred to is the baseline of our paper.

In Table 1, it is evident that the complexity algorithm outperforms the Distance algorithm in terms of accuracy. This may be attributed to the fact that the Distance algorithm fails to deliver certain solutions, which results in a lower overall performance.

By correcting the limitation of the Distance algorithm, it is hoped that an increase in accuracy will be obtained.

In the Sigmorphon Analogy Dataset, the provided answers for certain analogy puzzles do not encompass all potential solutions. This is illustrated by the following example.

$$\begin{aligned} &\textit{asked} : \textit{ask} :: \textit{seemed} : x \\ &\qquad\qquad \Rightarrow \quad x = \textit{seem} \text{ or } \textit{seme} \end{aligned}$$

The dataset suggests that the correct answer to this analogy is "seem". It does not consider "seme" as a solution. However, theoretically, "seme" satisfies the criteria for solving analogies adopted by the Distance algorithm and should be identified as a potential answer. The previous version of the Distance algorithm outputs only one of the solutions. Consequently, there is a need to propose an algorithm that would generate the two possible answers for this analogy, including the one that is not recognized by the dataset as a solution.

2. Proposal: Recursive version for the Distance algorithm

In order to deliver all possible solutions for an analogy puzzle, we first examine why the previous algorithm delivers only a single solution. Let us consider an example analogy puzzle:

$$aa : ab :: aaa : x$$
$$\Rightarrow \quad x = ?$$

The processing of the Distance algorithm involves several steps. Firstly, it checks whether the analogy puzzle satisfies a specific constraint, namely whether all the letters in the string aa appear either in the string ab or teh string aaa. Once this constraint is met, the algorithm proceeds to compute edit distance matrices [14] between the strings aa and ab, as well as between the strings aa and aaa. The result is illustrated below. The string aa appears as a vertical axis around which the matrices are built. Notice that, consequently, the string ab is written from right to left for symmetry reasons.

$$\begin{array}{cccccc} b & a & & a & a & a \\ \cdot & 0 & a & 0 & 1 & \cdot \\ 1 & \cdot & a & \cdot & 0 & 1 \end{array}$$

Then the algorithm computes the edit distance traces [15] in each of the edit distance matrices, so as to deliver an answer by establishing a correspondence between the two traces found.

$$\begin{array}{cccccc} b & a & & a & a & a \\ \cdot & \color{red}{0} & a & \color{red}{0} & \color{red}{1} & \cdot \\ \color{red}{1} & \cdot & a & \cdot & 0 & \color{red}{1} \end{array}$$

The established correspondence can be visualised in the following table, where actions that consist in outputting one character at a time in the solution, are taken according to the correspondence between the two traces, relying on the directions followed along the traces.

dir_{AB}	dir_{AC}	do
diagonal	diagonal	copy b
diagonal	horizontal	copy a
diagonal	diagonal	copy a

As a result, the algorithm delivers the solution aab. The algorithm explores the traces only once and stops. Consequently, it does not output other possible solutions: aab and aba.

To get the solution aba, the algorithm needs to follow the following steps, indicated in the following table, similar to the previous one above.

dir_{AB}	dir_{AC}	do
diagonal	horizontal	copy a
diagonal	diagonal	copy b
diagonal	diagonal	copy a

The traces that have been followed in this table, are other paths that allow to still get the minimal edit distances between aa and ab on the one hand, and between aa and aaa on the other hand. The edit distance matrices are the same matrices as above, but the traces are different. Here, in fact, only the trace between aa and aaa is different.

$$\begin{array}{cccccc} b & a & & a & a & a \\ \cdot & 0 & a & 0 & 1 & \cdot \\ 1 & \cdot & a & \cdot & 0 & 1 \end{array}$$

The new trace shown here is not computed by the algorithm, although it corresponds to a minimal edit distance between the strings involved in the analogy puzzle. The reason for the algorithm to provide only one solution is that it simply does not produce and explore all possible traces within the matrices.

To address this limitation, we propose to implement a recursive version of the Distance algorithm. By adopting a recursive approach, the algorithm will go beyond the initial trace it finds and backtrack to the beginning, thereby exploring alternative traces. This modification will allow the algorithm to consider multiple traces and generate a other solutions for the same analogy puzzle.

It is important to note that while the main idea of the algorithm remains the same, the proposed change lies in the implementation of the algorithm. The recursive version enhances the algorithm's capability to explore and generate a more comprehensive set of traces, thus improving the overall solution output for the analogy puzzle, that is its recall.

3. Generation of Data for the Experiments

We evaluate our work by performing experiments on the previously introduced Sigmorphon Analogy Dataset. In addition, we use automatically generated data, for which we control the number of solutions, so as to ensure that the new version of the Distance algorithm actually outputs the exact number of possible solutions for a given analogy puzzle and that the solutions are exact.

The Distance algorithm relies on a definition of analogy given in [10]. By noting the distance between two strings A and B by $d(A, B)$ and the count of a character a in a string A by $|A|_a$, this definition is as follows:

$$A : B :: C : D \quad \Rightarrow \begin{cases} d(A,B) & = d(C,D) \\ d(A,C) & = d(B,D) \\ |A|_a + |D|_a & = |B|_a + |C|_a, \forall a \end{cases} \tag{1}$$

In order to cross-check the validity of our new recursive version of the Distance algorithm, we choose to generate our additional test data based on another definition of analogy between strings, namely the one given in [4]. According to this definition, D is a solution to the analogy puzzle $A : B :: C : ?$ if D is a string that belongs to the shuffle string of the strings B and C, from which all the characters of string A have been discarded in the same order. If we denote with $B \bullet C$ the shuffle of strings B and C, and the discarding operation with $\setminus$, then this definition states:

$$D \in (B \bullet C) \setminus A \tag{2}$$

3.1. Analogy puzzles with no solution

Based on the observation at the foundation of the definition of analogy between strings by distance [10, p. 731], and similarly, based on the definition by shuffle [16, p. 123], we can state that, if an analogy has solutions, all the characters in A should appear in B and C at least once. Consequently, generating analogy puzzles which have no solution can be done by the following procedure which basically enforces a character in A to appear in neither B nor in C.

- Step 1: Randomly generate a string A.
- Step 2: Randomly select a character, delim, in A which will not appear in B or C.
- Step 3: Randomly generate strings B and C that do not contain the character selected in – Step 2:.

Below is an instance of the above generation process to get an analogy without any solution.

- Step 1: $A \leftarrow$ *abcd*.
- Step 2: delim $\leftarrow$ *a*.
- Step 3: $B, C \leftarrow$ *bc*, *efgc* (B and C do not contain delim $=$ *a*).

With these values, the following analogy puzzle has no solution:

$$\begin{aligned} abcd : bc :: efgc : x \\ \Rightarrow \quad x = \text{no solution} \end{aligned}$$

3.2. Analogy puzzles with only one solution

3.2.1. Analogy puzzles with only one solution, the empty string

As mentioned earlier, in the case of an analogy with solutions, every character present in string A must also appear in either string B or string C. Consequently, to create an analogy with only one solution that is the empty string, it suffices to create B and C from A by distributing each character in A either in B or in C, in the same order. As a result, all characters from A are found in B or C. In this setting, for the shuffle explanation, $B \bullet C = A \bullet D$ implies that D is the empty string. For the distance explanation, because of the counts of characters in A and D being equal to those in B and C, this also implies that D is the empty string.

A degenerated case of the above is to split A into two parts, the left and the right parts, i.e., B is a prefix of A and C is the remaining suffix of A. This makes the analogy $B.C : B :: C : \varepsilon$.

3.2.2. Analogy puzzles with only one solution, which is not the empty string

Drawing upon the approach of generating analogy puzzles with an empty string solution, it becomes feasible to incorporate additional sub-strings within both B and C. These sub-strings are concatenated in the sequential order of their insertion into B and C, ultimately forming the solution to the analogy puzzle, denoted as D.

- Divide A into a prefix and a suffix B' and C', i.e., $A = B'.C'$.
- Create additional sub-strings that do not share any character with A.
- Insert the additional sub-strings into B' or C'.

In the last step, the following constraints are crucial in guaranteeing the uniqueness of the solution:

- No additional sub-string should be added as a suffix of B';
- No additional sub-string should be added as a prefix of C'.

To justify these constraints, consider the following analogy puzzle:

$$\begin{aligned} &\textit{abcd} : \textit{ab} :: \textit{cd} : x \\ &\qquad\qquad \Rightarrow \quad x = ? \end{aligned}$$

Here, the prefix is $B' = ab$ and the suffix is $C' = cd$. Let us denote the additional sub-strings by M and N. We insert M into the middle of string B'. We insert N at the beginning of string C', which does not respect the constraint given above. Clearly, because of that, the obtained analogy puzzle has possibly several solutions:

$$\begin{aligned} &\textit{abcd} : \textit{aMb} :: \textit{Ncd} : x \\ &\qquad\qquad \Rightarrow \quad x = \textit{any string in } M \bullet N \end{aligned}$$

Hence, the procedure for generating analogy puzzles that possess a unique solution can be outlined as follows:

- Step 1: Randomly generate a string A and select a position to divide A into prefix B' and suffix C'.
- Step 2: Create any number of sub-strings randomly, each without any of the characters in A.
- Step 3: Insert the sub-strings into B' and C' and get B and C, respecting the constraints:
  - no sub-string is inserted as a suffix of the prefix B'.
  - no sub-string is inserted as a prefix of the suffix C'.

Here is a generation instance following the above procedure:

- Step 1: Generate a string $A = abcd$. Select the position of a character, for instance, *"c"* to divide A into prefix *abc* and suffix *d*.
- Step 2: Create three sub-strings *op*, *mn* and *opmn*.
- Step 3: Insert the sub-strings respecting the constraints. For instance, get $B = amnbopc$ and $C = dopmn$.

D is necessarily the concatenation of the sub-strings in the order they have been inserted in the prefix and suffix of A, which is unique. Hence, D is unique. As a result the obtained analogy puzzle has only one unique solution:

$$\begin{aligned} &\textit{abcd} : \textit{amnbopc} :: \textit{dopmn} : x \\ &\qquad\qquad \Rightarrow \quad x = \textit{mnopopmn} \end{aligned}$$

3.3. Analogy puzzles with several solutions

As explained in the section regarding the generation of analogy puzzles with only one solution, certain constraints serves as a means of ensuring solution uniqueness. However, in the absence of such constraints, alternative methodologies can be employed to generate analogy puzzles with multiple solutions.

There are primarily two constraints we previously mentioned, which are:

1. Position:
   No additional sub-string should be added as a suffix of B', and no additional sub-string should be added as a prefix of C'.

2. Character:
   Create additional sub-strings that do not share any character with A.

3.3.1. Without the position constraint

Let us consider a slight variation of the example given in section 3.2.2 that ignores the constraint on position:

$$\begin{aligned} &\textit{abcd} : \textit{abM} :: \textit{Ncd} : x \\ &\qquad\qquad \Rightarrow \quad x = \textit{any string in } M \bullet N \end{aligned}$$

In this analogy puzzle, we denote the additional sub-strings by M and N. We add M as the suffix of B' and N as the prefix of C', which has the consequence that the analogy puzzle has potentially several solutions.

Hence, a first possible procedure for generating analogy puzzles that possess several solutions can be outlined as follows:

- Step 1: Randomly generate a string A and select a position to divide A into prefix B' and suffix C'.

- Step 2: Create two sub-strings randomly, each without any of the characters in A. To ensure that $M.N$ contains two or more elements, we impose that each of the two sub-strings contains at least two different characters.

- Step 3: Add one sub-string as the suffix of B' and another as the prefix of C' to get B and C.

3.3.2. Without the character constraint

Here is an example that ignores the character constraint:

$$\begin{aligned} &abcd : aOb :: cMdNcMdN : x \\ &\quad \Rightarrow \quad x = \textit{any string in } OcMdNcMdN \setminus cd \end{aligned}$$

In this analogy puzzle, we denote the additional sub-strings by O, M, and N. We insert O into string B', insert M and N into string C' and repeat C' several times to get C. This creates an analogy puzzle with several solutions because of the multiple possibilities to erase cd from C.

Hence, a second procedure for generating analogy puzzles that possess several solutions can be outlined as follows:

- Step 1: Randomly generate a string A and select a position to divide A into prefix B' and suffix C'.
- Step 2: Create any number of sub-strings randomly, each without any of the characters in A.
- Step 3: Insert the sub-strings into B' and C' and get B and C'', respecting the constraints:
  - no sub-string is inserted as a suffix of the prefix B'.
  - no sub-string is inserted as a prefix of the suffix C'.
- Step 4: Repeat C'' any number of times to get C.

4. Experiments

We run both the previous version of the Distance algorithm and its new recursive version on the above-mentioned data sets, i.e., the Signmorphon ananlogy datatset and the datasets of analogy puzzles with zero, one only or several solutions. We measure their processing time, precision, recall, and F-measure.

The results given in Table 2 show that, while the recursive version may exhibit a longer processing time in average compared to the previous version, its recall is 100% on the automatically generated testsets, and higher (98.5%) than the previous version (92.2%) on the Sigmorphon Analogy Dataset. We conclude that the recursive version successfully delivers almost all of the solutions of the analogy puzzles contained in our datasets.

We also compare the results to the methods in [11], [4], and [10]. For that we add the results of the new recursive version on the Sigmophon Analogy data set to Table 1 to obtain Table 1.

The comparative analysis on the Sigmorphon Analogy Dataset reveals that, except for one language, Spanish, the new version of the Distance algorithm introduced in this paper consistently outperforms the three alternative methods. Thanks to its higher recall, this new version demonstrates superior performance.

This observation suggests that the relatively poorer performance of the previous Distance algorithm can be attributed to its failure to capture all the possible solutions for certain analogy

Algorithm	Dataset	Average time (μs)	Precision (%)	Recall (%)	F-measure
previous	Sigmorphon	1.40	92.0	92.2	92.0
recursive		565.00	34.8	98.5	51.2
previous	Zero solution	0.17	100.0	100.0	100.0
recursive		0.26	100.0	100.0	100.0
previous	One solution	0.63	97.2	81.9	88.9
recursive		1.38	99.7	100.0	99.8
previous	Several solutions	1.26	96.7	30.7	46.6
recursive		1.83	99.4	100.0	99.7

Table 2
Assessment of previous and recursive versions of the Distance algorithm on the four different data sets. Of more importance to us is the recall.

Language	Number of analogies	Method			
		Complexity [11]	Distance [10]	Shuffle [7]	Recursive
Arabic	165,113	87.18%	93.33%	81.91%	**98.91%**
Finnish	313,011	93.69%	92.76%	78.75%	**97.13%**
Georgian	3,066,273	99.35%	97.54%	88.42%	**99.85%**
German	730,427	98.84%	96.21%	95.42%	**99.81%**
Hungarian	2,912,310	95.71%	92.61%	86.02%	**98.62%**
Maltese	28,365	96.38%	84.72%	91.84%	**98.17%**
Navajo	321,473	81.21%	86.87%	78.95%	**97.45%**
Russian	552,423	96.41%	97.26%	95.46%	**99.48%**
Spanish	845,996	**96.73%**	96.13%	94.42%	96.19%
Turkish	245,721	89.45%	69.97%	70.06%	**98.63%**
Total	9,181,112	96.41%	94.34%	87.93%	**98.50%**

Table 3
Table 1 with the results obtained by the recursive version of the Distance algorithm proposed in this paper. Best results in boldface.

puzzles. The absence of these solutions significantly impacted the recall. Conversely, the new recursive algorithm addresses this limitation by delivering a more comprehensive set of solutions, resulting in a higher recall for almost all languages, and consequently a higher average recall.

5. Limitations: Precision

The precision of the new recursive algorithm on the Sigmorphon Analogy Dataset is not 100% due to the nature of the approach, which outputs multiple solutions in many cases, whereas the

Sigmorphon Analogy Dataset only expects a single solution.

For instance, consider the following analogy:

$$\begin{aligned} &\textit{fa}\underline{\textit{k}}\textit{urātu} : \textit{ustrāliyyātu} :: \textit{fa}\underline{\textit{k}}\textit{ūrun} : x \\ &\qquad\qquad \Rightarrow \quad x = \textit{ustrāliyyun} \text{ or } \textit{ustrliyyāun} \end{aligned}$$

According to the Sigmorphon Analogy Dataset, the expected answer for this analogy is *ustrāliyyun*. However, theoretically, the answer *ustrliyyāun* is also a possible solution. Although it satisfies the definition of analogy on which our algorithm is based, it is not considered a valid solution within this particular linguistic context.

As mentioned in the section discussing the production of the automatically generated analogy puzzles, these additional solutions can be seen as "noise" since they do not align with the general notion of what constitutes a solution. Evaluating the effectiveness of our approach becomes problematic when using only the Sigmorphon Analogy Dataset, as it primarily focuses on a single expected solution rather than capturing the full scope of potential solutions. Testing solely on the Sigmorphon Analogy Dataset may not provide a comprehensive evaluation of whether our goals have been achieved.

6. Conclusion

In this paper, we built upon an existing algorithm for solving analogies. Our objective was to make this algorithm deliver all solutions for an analogy puzzle when there are multiple solutions. To accomplish this, we introduced recursivity to systematically explore all possible edit distance traces in the representation of analogy puzzles by edit distance matrices. Thanks to this we are able to enumerate all possible solutions of an analogy puzzle.

To evaluate the effectiveness of our proposal, we generated a dataset comprising analogies with different characteristics, i.e., cases with no solution, cases with one solution, and cases with multiple solutions. We presented the methods adopted to automatically produce analogy puzzles in all these different cases. The generated dataset allowed us to conduct an analysis on specific cases and ascertain the ability of our recursive version of the algorithm to deliver all existing solutions. Experiments demonstrated that our proposed new version of the algorithm successfully achieves this goal.

To summarize, by introducing a recursive approach we could expand the scope in solutions and could increase the performance of the Distance algorithm on the dataset of analogy puzzles extracted from the Sigmorphon Analogy Dataset.

Acknowledgments

This paper has been partially supported by the JSPS project Kakenhi Kiban C n° 21K12038 entitled "Theoretically founded algorithms for the automatic production of analogy tests in Natural Language Pocessing".

References

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

[2] R. R. Hoffman, Monster analogies, AI magazine 16 (1995) 11–11.

[3] P. Langlais, Mapping source to target strings without alignment by analogical learning: A case study with transliteration, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2013, pp. 684–689.

[4] P. Langlais, P. Zweigenbaum, F. Yvon, 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 Association for Computational Linguistics (EACL 2009), Association for Computational Linguistics, Athens, Greece, 2009, pp. 487–495. URL: https://www.aclweb.org/anthology/E09-1056.

[5] R. Fam, Y. Lepage, S. Gojali, A. Purwarianti, Indonesian unseen words explained by form, morphology and distributional semantics at the same time, in: Proceedings of the 23rd Annual Meeting of the Japanese Association for Natural Language Processing (NLP 2017), Tsukuba, Japan, 2017, pp. 178–181.

[6] R. Fam, Y. Lepage, S. Gojali, A. Purwarianti, A study of explaining unseen words in Indonesian using analogical clusters, in: Proceedings of the 15th International Conference on Computer Applications (ICCA-17), Yangon, Myanmar, 2017, pp. 416–421.

[7] P. Langlais, A. Patry, Translating unknown words by analogical learning, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 877–886.

[8] M. Nagao, A framework of a mechanical translation between japanese and english by analogy principle, Artificial and human intelligence (1984) 351–354.

[9] Y. Lepage, E. Denoual, Purest ever example-based machine translation: Detailed presentation and assessment, Machine Translation 19 (2005) 251–282.

[10] Y. Lepage, Solving analogies on words: an algorithm, in: Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'98), volume I, Montréal, 1998, pp. 728–735. doi:10.3115/980845.980967.

[11] P.-A. Murena, M. Al-Ghossein, J.-L. Dessalles, A. Cornuéjols, et al., Solving analogies on words based on minimal complexity transformation., in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2020, pp. 1848–1854.

[12] Y. Lepage, Analogy and formal languages, Electronic Notes in Theoretical Computer Science 53 (2004) 180–191. URL: https://www.sciencedirect.com/science/article/pii/S1571066105825824. doi:https://doi.org/10.1016/S1571-0661(05)82582-4, proceedings of the joint meeting of the 6th Conference on Formal Grammar and the 7th Conference on Mathematics of Language.

[13] Y. Lepage, Character-position arithmetic for analogy questions between word forms., in: Proceedings of the International Conference on Case-Based Reasonong (ICCBR) (Workshops), 2017, pp. 23–32.

[14] E. Ukkonen, Algorithms for approximate string matching, Information and control 64 (1985) 100–118.

[15] R. A. Wagner, M. J. Fischer, The string-to-string correction problem, Journal of the ACM (JACM) 21 (1974) 168–173.

[16] N. Stroppa, F. Yvon, An analogical learner for morphological analysis, in: Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL 2005), Ann Arbor, MI, 2005, pp. 120–127.

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

Weihao Mao[1,*], Yves Lepage[1]

[1]Graduate 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)

CEUR-WS.org/Vol-3438/paper_02.pdf

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:

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

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

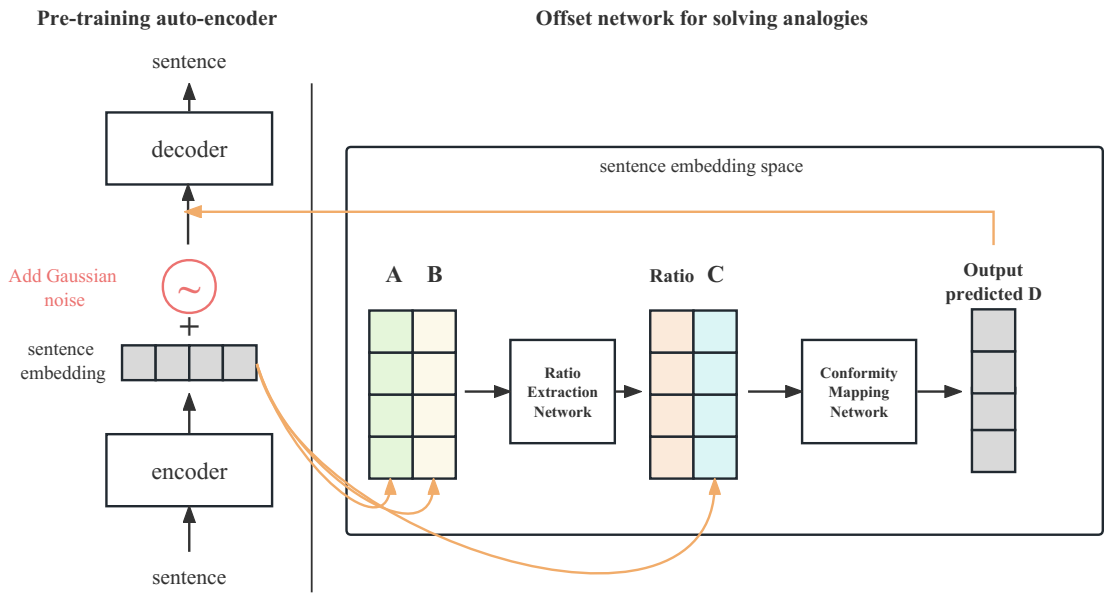


Figure 1: The diagram of the embedding-to-embedding method for solving sentence analogies

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 Tatoeba[1] 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**

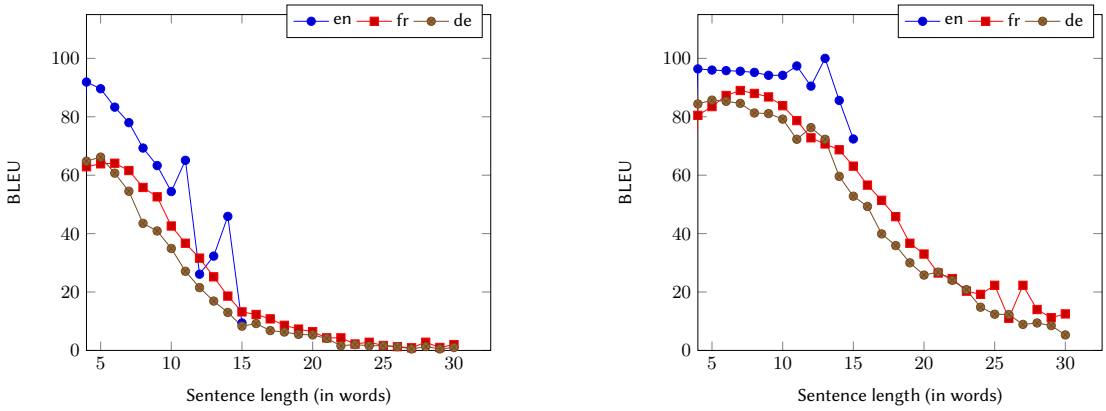
We conducted experiments to solve sentence analogies using different combinations of methods, i.e.,

- *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%.



(a) simple summation (b) encoder of autoencoder

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 generates 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

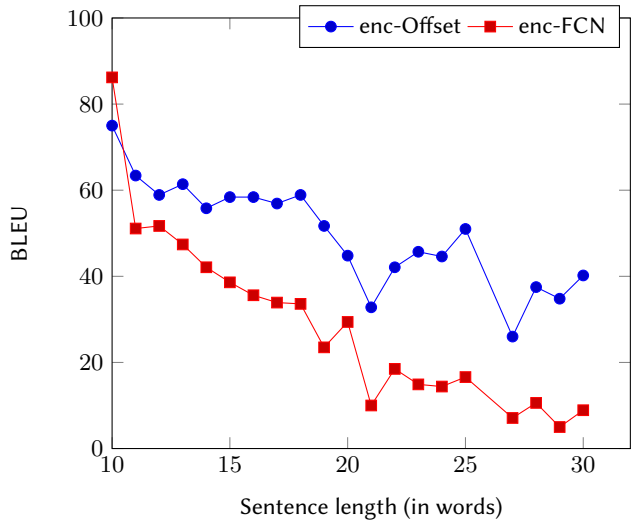


Figure 3: Performance of models on sentences with different lengths in French

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. Alsaidi, 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.

Less is Better: An Energy-Based Approach to Case Base Competence

Esteban Marquer[4,*,†], Fadi Badra[1,*,†], Marie - Jeanne Lesot[3,*,†], Miguel Couceiro[4,*,†] and David Leake[2,*,†]

[1]*Université Sorbonne Paris Nord, Laboratoire d'Informatique Médicale et d'Ingénierie des Connaissances en e-Santé, LIMICS, Sorbonne Université, INSERM, F-93000, Bobigny, France*

[2]*The Luddy School of Informatics, Computing, and Engineering, Indiana University, USA*

[3]*Sorbonne Université, CNRS, LIP6, F-75005 Paris, France*

[4] *University of Lorraine, CNRS, Loria, Nancy, France*

Abstract

This paper revisits the notion of case base competence in the light of recent advances in the modeling of analogical reasoning, based on the idea of similarity transfer from a situation space to an outcome space. For that we consider the CoAT indicator, that measures the compatibility between two similarity measures on a case base, and use it to define an intrinsic measure of competence of a case base with respect to a reference set. Initial experimental results show that the proposed competence measure correlates with the performance of the CoAT prediction algorithm. In fact, our preliminary results seem to indicate that, under some initial conditions, our competence based model can fit any classification boundary. We then revisit the notions of case competence and locality, and show that some source cases may degrade the overall case base competence while others may improve it, and that a given source case may have disparate influence on different regions of the case space.

Keywords

Competence models, case base compression, energy-based models, case base maintenance, case-based classification

1. Introduction

Case bases are one of the main sources of knowledge used in case-based reasoning (CBR), along with similarity knowledge, adaptation knowledge and domain knowledge [1]. Case acquisition and maintenance therefore constitute crucial steps in the knowledge engineering process of a CBR system. Acquiring and maintaining cases may be expensive, and case storage capacity

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.

†These authors contributed equally.

esteban.marquer@loria.fr (E. Marquer); badra@sorbonne-paris-nord.fr (F. Badra); Marie-Jeanne.Lesot@lip6.fr (M. -. J. Lesot); miguel.couceiro@loria.fr (M. Couceiro); leake@indiana.edu (D. Leake)

https://emarquer.github.io/ (E. Marquer); https://limics.fr (F. Badra); https://lip6.fr (M. -. J. Lesot); https://members.loria.fr/mcouceiro/ (M. Couceiro); https://homes.luddy.indiana.edu/leake/ (D. Leake)

0000-0003-2315-7732 (E. Marquer); 0000-0002-2437-8230 (F. Badra); 0000-0002-3604-6647 (M. -. J. Lesot); 0000-0003-2316-7623 (M. Couceiro); 0000-0002-8666-3416 (D. Leake)

© 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)

CEUR-WS.org/Vol-3438/paper_03.pdf

may be limited or constrained by selection criteria. Crafting a case base for a given task thus requires addressing questions such as "which cases should be included in the case base?", which can alternatively be expressed as "which cases are the most competent?", where the definition of the competence notion can be seen as the formalization of this issue.

Such questions have been extensively studied in the literature on case base maintenance (*e.g.,* [2, 3, 4, 5, 6, 7, 8]). Most competence models assume that problems are solved by a k-Nearest Neighbor algorithm (often with $k = 1$) augmented by case adaptation, and are strongly influenced by the way this algorithm works. For instance, it is often assumed that only the most similar cases may contribute to solving a new case, provided that they are themselves adaptable. The competence of a case is typically assessed by computing its coverage, *i.e.*, the set of cases that it may contribute to solving. Yet beyond approaches based on the k-Nearest Neighbor algorithm, no case-based prediction algorithm actually complies with this assumption. Algorithms such as CCBI [9], PossIBL [9], or CoAT [10], take into account the similarities with *all* source cases in order to make a prediction. Here we examine competence criteria suitable for such approaches.

Competence methods yield guidance for case deletion to minimize the competence loss in the case base compression process [5, 6]. However, determining a suitable notion of competence over the whole case base is challenging and, to our knowledge, no theoretical guarantees exist to relate the competence of a case base and the performance of a corresponding CBR system.

Recent advances in the modeling of analogical transfer shed new light on the case base competence problem. It has been shown [11] that all case-based prediction methods share the common inference principle based on the transfer of similarity knowledge from the situation space, in which the cases are described, to an outcome space, in which their attached solutions are described. This transfer can then be achieved by optimizing a measure of compatibility between two similarity measures, respectively associated with each of the two spaces. The latter idea motivated the case-based prediction method CoAT [10, 12, 13] that relies on the optimization of a global compatibility indicator between two similarity measures on the case base. In this framework, the predicted outcome is the one that entails the least increase in the value of the CoAT indicator. The latter can be interpreted as an intrinsic indicator of the optimality of the case-based setting for the task at hand. Preliminary results showed it can be used to assess the quality of similarity measures or of solutions.

In this paper, we further explore this indicator to address the problem of case competence. The main idea is to exploit this indicator to define an intrinsic measure of competence of a case base, which could be used later to obtain theoretical guarantees on the link between the competence of a case and the performance of a case-based classifier or predictor. To do so, we first propose to interpret the CoAT global indicator in an energy-based framework [14]. Then the competence of a source case can be intuitively related to its ability to reduce the energy of correct outcomes and to increase the energy of incorrect outcomes. For instance, in a classification setting, this would mean the case is lowering the energy of the good class, and increasing the energy of all others.

This new approach to the problem of case (base) competence has two noteworthy consequences. First, rather than taking the traditional case base maintenance view of considering only the nearest cases, it considers the compatibility of all cases in the case base. Intuitively, this could be important for deletion scenarios, because case bases with lower energy should

provide more stable results when cases are deleted. Second, this makes clear the potential ramification that case deletion could either decrease or increase system performance: cases may be competent (increased overall performance) w.r.t. a given class, while entailing competence degradation (decreased overall performance) w.r.t. another class.

To support the latter and establish the relation between competence and performance of a case-based system, we propose two loss functions (see Sec. 4): one that corresponds to the intuitive notion of competence (*i.e.*, counting positively the energy of correct outcomes and negatively the energy of incorrect ones), and the other inspired by the hinge loss. We perform a comparative study of the two and observe that the latter is preferable to the former. Thus focusing on the hinge loss, we conduct several empirical studies to assess both the performance and robustness of this CoAT-based competence notion (see Sec. 5) in various initial settings. These experiments also indicate the potential use of our competence notion for case base compression and maintenance purposes. Furthermore, they support that our competence-based framework can produce surrogate models capable of approximating different classification boundaries.

The paper is organized as follows. In Sec. 2 we briefly discuss well-known approaches to case base maintenance, and recall key definitions from related work on case competence. The definition of the CoAT indicator is recapped in Sec. 3 and then used in Sec. 4 to propose a new definition of case base competence and of competence of an individual case. We present several empirical studies in Sec. 5 to support our performance and robustness claims, and to analyze the behavior of our approach both quantitatively and qualitatively. Sec. 6 concludes the paper and discusses several perspectives for future work.

Main contributions. The main contributions of the paper are the following:

- We introduce an energy-based framework that relies on the optimization of the CoAT indicator as a measure of similarity compatibility, and which is used to propose new measures of case base competence w.r.t. different loss functions.
- We show empirically that thus defined, the case base competence is tightly linked to the performance of case-based models, which constitutes a promising step towards theoretical guarantees of performance.
- We propose fine-grained competence notions, namely, w.r.t. individual source cases and w.r.t. individual reference cases, which can be used to identify areas of "expertise" of cases in a case-based system.
- We present an empirical study to assess the robustness of the proposed approach w.r.t. to different case base initializations and reference case sets, followed by an iterative and qualitative analysis that shows the potential of the proposed approach for case base maintenance and compression, as well as for fitting any classification boundary.

2. Basic Background and Motivation

This section briefly presents the notation used throughout the paper and recaps the definition of the case base maintenance task in the CBR setting.

2.1. Key Definitions

Let $\mathcal{S}$ denote an input space, and $\mathcal{R}$ an output space. An element of $\mathcal{S}$ is called a *situation*, and an element of $\mathcal{R}$ is called an *outcome*, or a result. A set $CB = \{(s_1, r_1), \ldots, (s_n, r_n)\}$ of elements in $\mathcal{S} \times \mathcal{R}$ is called a *case base*. An element $c = (s, r) \in CB$ is called a *source case*. In addition, the spaces $\mathcal{S}$ and $\mathcal{R}$ are respectively equipped with the similarity measures $\sigma_{\mathcal{S}}$ and $\sigma_{\mathcal{R}}$, that respectively denote the similarity measure on situations and on outcomes. Let $\mathcal{T} \subset \mathcal{S} \times \mathcal{R}$ be a set of cases called a *reference set*, and $c_t = (s_t, r_t) \in \mathcal{T}$ be a reference case. We will write $(s_t, \hat{r})$ to denote a potential case constructed by keeping the same situation $s_t \in \mathcal{S}$, but choosing a different outcome $\hat{r} \in \mathcal{R}, \hat{r} \neq r_t$ for the case.

2.2. Case Base Maintenance

Case base maintenance revises the contents or organization of a case base to improve performance, and is a longstanding research area for CBR (*e.g.*, [15]). Much of this work has studied case base compression by case deletion [5]. Compression efforts were initially motivated by the desire to control retrieval costs and respect storage constraints. Advances in computational power have reduced some of these concerns in practice [16] but compression remains useful when efficiency is paramount and for reasons such as reducing the number of cases for a knowledge engineer to maintain. Deletion of cases may remove the knowledge required to solve particular problems, motivating maintenance work focused on retention of case base competence, which Smyth [6] and McKenna define as the range of problems a CBR system can successfully solve.

Case base compression strategies are often deletion-based, aimed at successively removing cases whose loss will least harm competence. Estimates of case competence contributions are commonly done based on the existing cases in the case base, under the representativeness assumption [6] that the case base is a good predictor of the distribution of future problems. This assumption is expected to hold for domains well suited to CBR, when the case base is sufficiently mature, though may be endangered by problem drift (*e.g.*, [17]). Estimation of expected case competence contributions is commonly based on considering relationships between cases and their nearest neighbors in the case base, favoring cases that have high coverage of other cases and low reachability, *i.e.*, that are recoverable from fewer cases [6].

This paper presents a maintenance perspective that is novel in three ways. First, rather than emphasizing the relationship of cases to nearby neighbors, the core of the approach is a global optimization of a case base energy function. Second, rather than using a global approximation of future problems, it defines competence with respect to specific reference sets. Third, it questions the assumption that case base compression entails competence loss and illustrates that compression may actually enhance performance, providing a new motivation for case base compression.

3. The CoAT Method

The CoAT method [10, 12, 13] performs analogical transfer by minimizing a global indicator of compatibility between two similarity measures. In this section, we recall the definition of this

indicator, and show that it can be seen as an energy function.

3.1. Definition of the CoAT Indicator

The compatibility of $\sigma_{\mathcal{R}}$ with $\sigma_{\mathcal{S}}$ for a given case base CB is measured globally on the case base CB, by a global indicator denoted $\Gamma(\sigma_{\mathcal{S}}, \sigma_{\mathcal{R}}, CB)$. The latter takes an ordinal point of view on the whole case base CB, by checking if the order induced by $\sigma_{\mathcal{R}}$ is the same as the one induced by $\sigma_{\mathcal{S}}$. The following continuity constraint is tested on each triple of cases (c_0, c_i, c_j), with $c_0 = (s_0, r_0)$, $c_i = (s_i, r_i)$, and $c_j = (s_j, r_j)$:

$$\text{if } \sigma_{\mathcal{S}}(s_0, s_i) \geq \sigma_{\mathcal{S}}(s_0, s_j), \text{ then } \sigma_{\mathcal{R}}(r_0, r_i) \geq \sigma_{\mathcal{R}}(r_0, r_j). \tag{C}$$

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

$$\begin{aligned}\Gamma(\sigma_{\mathcal{S}}, \sigma_{\mathcal{R}}, CB) = |\{((s_0, r_0), (s_i, r_i), (s_j, r_j)) \in CB \times CB \times CB \text{ such that} \\ \sigma_{\mathcal{S}}(s_0, s_i) \geq \sigma_{\mathcal{S}}(s_0, s_j) \text{ and } \sigma_{\mathcal{R}}(r_0, r_i) < \sigma_{\mathcal{R}}(r_0, r_j)\}|.\end{aligned}$$

For a new situation s_t, the transfer inference consists in finding the outcome r_t that leads to the new case $c_t = (s_t, r_t)$ that minimizes the value of the indicator:

$$r_t = \arg\min_{r \in \mathcal{R}} \Gamma(\sigma_{\mathcal{S}}, \sigma_{\mathcal{R}}, CB \cup \{(s_t, r)\}). \tag{1}$$

An important aspect to notice is that the CoAT method makes use of the whole case base CB, and not only the most similar case(s), in order to predict the outcome of the new case.

3.2. An Energy Function View on the CoAT Method

After briefly reminding the principles of the energy-based framework for solving machine learning tasks, this section proposes to interpret the CoAT optimization of the global indicator in this setting.

Energy-based models. Inspired from statistical physics, energy-based models [14] specify a probability distribution $p(x; \theta) = e^{-E_\theta(x)} / \int e^{-E_\theta(x)} dx$ via a parameterized scalar-valued function $E_\theta(x)$ called an *energy function*. In its conditional version, the definition of an energy function $E_\theta : \mathcal{X} \times \mathcal{Y} \longrightarrow \mathbb{R}$ associates to each pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ a scalar value $E_\theta(x, y)$ that represents the compatibility between the input x and the output y under the set of parameters θ. The energy function E_θ takes low values when y is compatible with x, and higher values when y and x are less compatible. The goal of the energy-based *inference* is to find, among a set of outputs $\mathcal{Y}$, the output $y^* \in \mathcal{Y}$ that minimizes the value of the energy function:

$$y^* = \arg\min_{y \in \mathcal{Y}} E_\theta(x, y).$$

Given a family of energy functions $E_\theta(x, y)$ indexed by a set of parameters θ, the goal of the *learning* step is to optimize the θ parameters in order to "push down" (*i.e.*, assign lower energy values to) the points on the energy surface that are around the training samples, and to "pull up" all other points. Contrastive divergence [18] is a common learning strategy that, given a numerical hyperparameter λ, consists in optimizing a contrastive loss function such as the hinge loss, which is defined, for a training sample (x_k, y_k) and a generated out of distribution sample $(x_k, \hat{y})$ by: $\ell(\theta, x_k, y_k, \hat{y}) = \max(0, \lambda + E_\theta(x_k, y_k) - E_\theta(x_k, \hat{y}))$. The hinge loss associates a loss value to a training sample (x_k, y_k) whenever its energy is not lower by at least a margin λ than the energy of the incorrect sample $(x_k, \hat{y})$.

The CoAT indicator as an energy function. The CoAT case-based prediction method can be interpreted in the energy-based model framework, in which the energy $E_\theta(s_t, r)$ of any new case (s_t, r) is given by the value of the Γ indicator when the case is added to the case base, *i.e.*,

$$E_\theta(s_t, r) = \Gamma(\sigma_\mathcal{S}, \sigma_\mathcal{R}, CB \cup \{(s_t, r)\}).$$

The input space $\mathcal{X}$ is the situation space $\mathcal{S}$ and the output space $\mathcal{Y}$ is the outcome space $\mathcal{R}$. The energy function $E_\theta : \mathcal{S} \times \mathcal{R} \longrightarrow \mathbb{R}$ measures the compatibility of the outcome similarities with the added situation similarities when the potential new case $\hat{c}_t = (s_t, r)$ is added to the case base. The energy function E_θ is parameterized by $\theta = (\sigma_\mathcal{S}, \sigma_\mathcal{R}, CB)$ which includes the case base CB. The goal of the energy-based *inference* is to find, among the set of potential outcomes $r \in \mathcal{R}$, the outcome r_t that minimizes the value of the energy function, and (1) can be reformulated as:

$$r_t = \arg\min_{r \in \mathcal{R}} E_\theta(s_t, r).$$

4. Measuring Competence

In this section we introduce new case (base) competence measures using the previous energy-based framework of the CoAT indicator, w.r.t. different loss functions. We then propose fine-grained variants of competence, at individual and reference case levels.

4.1. Idea of the Method

In the CoAT energy-based model, the energy function $E_\theta(s_t, r)$ is used to compute a (scalar) energy value for each potential outcome r of the new case c_t. The difference between the energy of the predicted outcome and the lowest energy of all other outcomes can be interpreted as a measure of prediction confidence. Therefore, our goal is to capture the idea that the competence of a case base should be related to its ability to maximize the prediction confidence, by decreasing the energy of the correct outcome of a new case and increasing the energy of incorrect outcomes.

We consider two different loss functions of the underlying energy-based model that take as input, besides the $\theta = (\sigma_\mathcal{S}, \sigma_\mathcal{R}, CB)$ parameters of the energy function, that are considered to be fixed, an auxiliary set of reference cases $\mathcal{T}$. The intuition is that, if $\sigma_\mathcal{S}$ and $\sigma_\mathcal{R}$ are fixed, optimizing such a loss function should allow us to learn the right case base CB for the task, *i.e.*, address the case base maintenance issue.

4.2. Competence of a Case Base

This section discusses two definitions of the competence of a case base CB with respect to a reference set $\mathcal{T}$, that are defined from two different loss functions of the energy-based model.

MCE loss competence. The first definition of competence we propose, denoted C_{MCE} relies on the notion of the minimum classification error loss ℓ_{MCE} [14] classically used in the energy-based framework. More precisely, C_{MCE} computes the average value, across the reference set, of this loss ℓ_{MCE} that is defined as the difference between the energy of the correct outcome and the minimum energy of a reference case if it were assigned a different outcome:

$$C_{MCE}(CB, \mathcal{T}) \quad = \quad -\frac{1}{|\mathcal{T}|} \sum_{c_t \in \mathcal{T}} \ell_{MCE}(CB, c_t),$$

where $\ell_{MCE}(CB, c_t) = E_\theta(s_t, r_t) - \min_{\hat{r} \neq r_t} E_\theta(s_t, \hat{r})$. For a correctly classified instance, $\ell_{MCE}(CB, c_t)$ is a negative value whose magnitude can be interpreted as the prediction confidence of CoAT, as mentioned previously. For an incorrectly classified instance, $\ell_{MCE}(CB, c_t)$ is a positive value that allows to measure the extent of the error, *i.e.*, how much the true class is missed. As a consequence, the lower the $\ell_{MCE}(CB, c_t)$ value, the better and, due to the - sign in the definition of $C_{MCE}(CB, \mathcal{T})$, the greater the $C_{MCE}(CB, \mathcal{T})$, the better, *i.e.*, the more competent CB is w.r.t. $\mathcal{T}$.

Hinge loss competence. The hinge loss competence C_{hinge} modifies the minimum classification error loss by integrating an additional parameter, denoted by λ, that corresponds to a margin. The values of $\ell_{MCE}(CB, c_t)$ that are lower than $-\lambda$ (corresponding to the instances with high prediction confidence) are not taken into account and not allowed to compensate for the misclassified instances:

$$C_{hinge}(CB, \mathcal{T}) \quad = \quad -\frac{1}{|\mathcal{T}|} \sum_{c_t \in \mathcal{T}} \ell_{hinge}(CB, c_t),$$

where $\ell_{hinge}(CB, c_t) = \max(0, \lambda + \ell_{MCE}(CB, c_t))$.

Comparison between the competence metrics. C_{MCE} is close to a direct translation of the notion of competence described in Subsec. 4.1. However, in C_{MCE}, the negative contributions to competence (incorrect predictions) and the positive ones (correct predictions) can cancel each other out. In other words, a high increase of the confidence for correctly predicted class can compensate for a lot of small misclassifications. This scaling issue between negative and positive contributions to C_{MCE} is avoided in C_{hinge} as only the negative contributions (to a margin) are accounted for.

4.3. Fine-Grained Competence: Case Level and Expertise Areas

This section proposes to break down the case base competence at a more refined level, considering the individual source and reference cases levels.

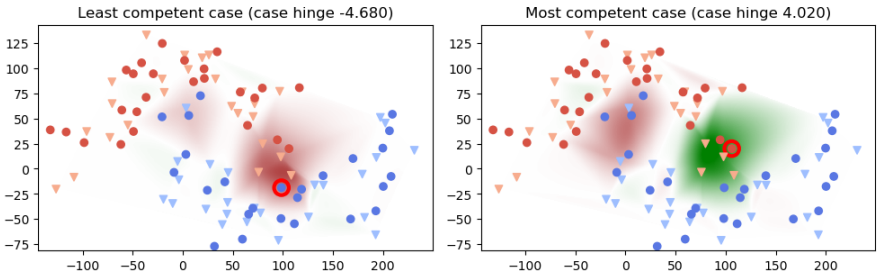


Figure 1: Influence map of 2 source cases c_1 and c_2 (circled in red) of the Half Moon dataset (CB=colored disks, $\mathcal{T}$=pale colored triangles): the background color shows, at each position x, y, the value of $influence(c_1, (x, y))$, where green corresponds to a positive value and red to a negative one.

Proposed definitions. We first propose to define the competence of a source case $c = (s, r) \in CB$ w.r.t. a reference set $\mathcal{T}$ as the loss of competence that would happen if this source case was deleted from the case base:

$$C(c, CB, \mathcal{T}) = C(CB, \mathcal{T}) - C(CB \setminus \{c\}, \mathcal{T}).$$

As for any competence measure, the greater, the better.

At an even finer level, we define the notion of competence locally as the contribution of a source case $c = (s, r) \in CB$ on each individual reference case $c_t \in \mathcal{T}$. Indeed, the competence $C(c, CB, \mathcal{T})$ over the reference set $\mathcal{T}$ equals the sum over $\mathcal{T}$ of the loss (*e.g.*, ℓ_{hinge} or ℓ_{MCE}): the above-defined competence of a case $c \in CB$ w.r.t. $\mathcal{T}$ can be expressed as

$$C(c, CB, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{c_t \in \mathcal{T}} influence_{CB}(c, c_t)$$

where $influence_{CB}(c, c_t) = \ell(CB, c_t) - \ell(CB \setminus \{c\}, c_t)$. This notion of case influence entails the idea of locality: all source cases contribute to the competence of the case base, but each source case may contribute differently on different regions of space. This enables the identification of *regions of expertise* of a source case. Since the loss distinguishes between correct and incorrect classification, case influence can also be used to identify those regions where a source case can improve the performance from those where performance is degraded.

Illustrative example. Figure 1 offers a visualization of the case competence and influence considering two source cases, circled in red, of the Half Moon dataset (see Subsec. 5.1), and the map of their influence values. The left figure shows the least competent source case ($C(c, CB, \mathcal{T}) = -4.680$) which degrades the performance on many reference cases (dark red regions) and contributes negatively to the overall competence. This case would be the first one to be removed in a case deletion strategy. The right figure shows the most competent case ($C(c, CB, \mathcal{T}) = 4.020$), which contributes positively to the competence of the case base: it improves the performance of a large set of reference cases (green regions). Interestingly, this case also harms the performance for some references of the opposing class.

Algorithm 1 Case deletion procedure

Require: An initial case base CB and a reference set $\mathcal{T}$
  while $|CB| > 0$ **do**
    $c_{worse} = \arg\min_{c \in CB} C(c, CB, \mathcal{T})$
    $CB = CB \setminus \{c_{worse}\}$
  end while

Case deletion procedure. Source case competence can be applied in a case deletion procedure, as described by Algorithm 1: at each iteration, the source case c_{worse} that contributes least to the competence of the case base CB w.r.t. the reference set $\mathcal{T}$ is deleted from the case base. To observe the effects of successive deletions this algorithm is exhaustive, but in practice deletion would repeat only until a stopping criterion is reached (*e.g.*, desired compression).

5. Experiments

We investigate experimentally the properties of the proposed case deletion procedure and competence definitions, in particular examining their correlation with the classification performance of the CoAT prediction algorithm. We also provide a stability analysis, as well as a qualitative analysis of the results.

5.1. Considered Artificial Datasets

The experiments are performed in a binary classification setting with three synthetic 2D datasets generated from 3 distributions coined Line, Ring, and Half Moon and respectively illustrated in Fig. 2a, 2b, and 2c.

The Line data are drawn from a uniform distribution defined on $[0, 2] \times [0, 3]$. They are labeled according to the arbitrary chosen line $f(x) = -x + 2.5$, and noise is added by randomly switching the label, with a probability of 20%, for cases within a 0.3 distance to the boundary.

For the Ring data, two classes are defined a concentric rings of radii 25 and 50. For each class, points are randomly sampled using polar coordinates, drawing the angle from a uniform distribution on $[0, 2\pi]$ and radius from a normal distribution $\mathcal{N}(\mu = r_c, \sigma = 10)$, where $r_c \in \{25, 50\}$ is the radius of the class. The theoretical decision boundary for the Ring data is the circle of radius 32.5.

The Half Moon dataset is generated with "make_moons" function from the Scikit-Learn library[1] with a noise of 0.2. The distribution is composed of two halves of a circle, one of which is shifted laterally by the radius. Each half-circle corresponds to a class.

In all three cases, $\mathcal{S} = \mathbb{R}^2$ and the associated similarity is a decreasing function of the standard Euclidean distance $\sigma_{\mathcal{S}}(x, y) = \exp(-d^2(x, y))$; the outcome space is $\mathcal{R} = \{0, 1\}$ equipped with $\sigma_{\mathcal{R}}(r_x, r_y) = 1$ if $r_x = r_y$ and 0, otherwise. Note that the three data distributions are more or less compatible with $\sigma_{\mathcal{S}}$ due to their geometry. In that regard, the limitations of $\sigma_{\mathcal{S}}$ help understand the performance of our approach when the similarity is not as good as it could be.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html

5.2. Experimental Protocol

For each of the three data distributions considered, we generate 1000 samples that we split into 20 non-overlapping subsets of 50 cases, each one being balanced in terms of classes. We separate them in 2 groups: 10 serve as initial case bases $CB_1, \ldots, CB_{10}$ and the others as reference sets $\mathcal{T}_1, \ldots, \mathcal{T}_{10}$. Fig. 2a, 2b, and 2c display the overall 1000 samples together with their label, showing in light colors the reference sets.

For each pair $(CB_i, \mathcal{T}_j)$, we apply the proposed compression algorithm. After each removal step, the classification results obtained by the CoAT algorithm applied with the current CB are assessed by the macro F1 on all reference cases $\bigcup_{k=1..10} \mathcal{T}_k$.

Fig. 2d, 2e and 2f show the evolution of this macro F1 criterion during the case deletion procedure, comparing the two proposed competence measures $C_{MCE}(c, CB, \mathcal{T})$ and $C_{hinge}(c, CB, \mathcal{T})$; the shade corresponding to the 95% confidence interval over the 100 combinations of initial case base and references. In Fig. 2g, 2h, and 2i, each line shows the results for one of the 10 case bases and the shades show the 95% confidence interval over the 10 reference sets. Reciprocally, in Fig. 2j, 2k, and 2l, each line corresponds to a reference set and the shades correspond to the 95% confidence interval over the 10 initial case bases.

5.3. Results

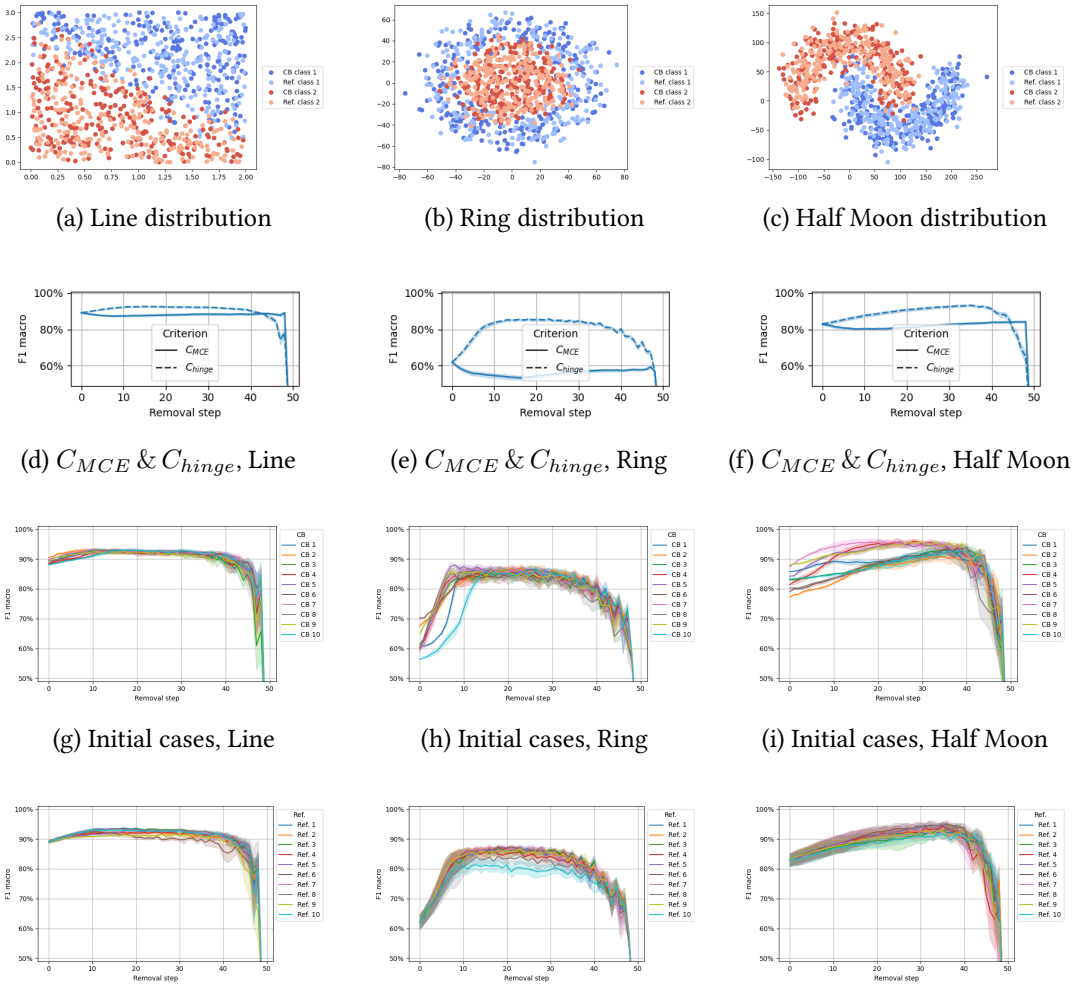
We study the behavior of the case-based models resulting from compression by performing both quantitative and qualitative analyses.

5.3.1. Competence Definitions and Correlation with Performance

In the second row of Fig. 2, we compare the evolution of the macro F1 when using either C_{MCE} or C_{hinge} for case competence in the compression process. With C_{MCE}, F1 remains at its maximum slightly longer, so a few more cases can be removed. However, with C_{MCE}, F1 remains at its initial value throughout the process and does not reach as high values as with C_{hinge}. For instance, on Ring, F1 reaches a value close to 60% for C_{MCE} and 85% for C_{hinge}.

Complementary experiments, whose curves are omitted for brevity, examine the evolution of the case base competence during the compression process. They show that, as desired, the case base competence remains constant or increases during compression when using C_{hinge}. On the other hand, using C_{MCE} causes an increasingly faster decrease of competence, making it a poor choice for case base compression. Also, by comparing the decrease when using C_{MCE} with F1, which is almost constant, it becomes striking that C_{MCE} is not directly correlated with predictive performance, and this is problematic in our vision of competence. As mentioned in Subsec. 4.2, prediction successes and failures are considered at the same time in C_{MCE}, but higher C_{MCE} could be an expression of higher confidence in already well predicted cases, of fewer errors, or of less confident errors. In that regard, C_{hinge} is more suitable as a competence measure as it measures how confident the model is in its errors, and thus higher C_{hinge} corresponds to fewer or less confident errors, which directly translates to higher performance.

The experiments described hereafter consider only C_{hinge}, as it is more interesting in terms of performance, stability across datasets, and is a better fit for the notion of competence.



(a) Line distribution (b) Ring distribution (c) Half Moon distribution

(d) C_{MCE} & C_{hinge}, Line (e) C_{MCE} & C_{hinge}, Ring (f) C_{MCE} & C_{hinge}, Half Moon

(g) Initial cases, Line (h) Initial cases, Ring (i) Initial cases, Half Moon

(j) Reference cases, Line (k) Reference cases, Ring (l) Reference cases, Half Moon

Figure 2: Evolution of the macro F1 (on all 500 reference cases) during the case deletion procedure, for Line, Ring and Half Moon. The distribution of the 1000 cases used is displayed in first row. In the second row, performance with C_{MCE} is compared with C_{hinge} (C_{MCE} & C_{hinge}). The performance with C_{hinge} is also detailed when grouping by case base initialization (third row) and reference set (fourth row).

5.3.2. Competence of the Case and Impact on Performance

Looking at the F1 evolution (see Fig. 2g to 2l) shows that no matter the initial cases in the case base or the reference cases used, the same trend of performance can be observed: *(i)* a raise, *(ii)* a plateau, and finally *(iii)* a faster and faster decrease. As our algorithm removes the cases by order of increasing competence, it appears that *(i)* corresponds to incompetent cases, *(ii)* to cases that are neither competent nor incompetent, and *(iii)* to competent cases. Going further, during phase *(i)* removing cases improves the performance, meaning that the removed cases were "polluting" the case base. In *(ii)*, the removed cases neither harm nor benefit the performance of

the case base, as such they can be considered redundant w.r.t. the remaining cases. The cases that remain are the most competent and useful ones, and in *(iii)* they are removed by order of increasing competence, leading to sharper and sharper drops in performance.

This behavior is similar to the footprint deletion procedure from Smyth and Keane [5], with the auxiliary, spanning, and support cases removed in *(ii)*, and pivotal cases removed in *(iii)*. Compared to [5], our procedure is more powerful as it can handle case bases that do not properly fit the distribution of the data, as harmful cases are removed in priority in *(i)*.

As we observe a striking parallel between the performance change and the compression step, it appears that C_{hinge} suits the intuition of competence, since the step at which a case is removed during compression is proportional to its competence. Furthermore, this general trend provides empirical guarantees that the maximum performance is reached just before the first significant decrease in performance, meaning we can stop the process as soon as we detect such a decrease.

5.3.3. Robustness of the Compression

Robustness w.r.t. the initialization of the case base. Fig. 2g, 2h, and 2i show that the initial cases in the case base change the initial performance and time needed to converge to the general trend of performance. In extreme cases of poor initial performance, the convergence might be delayed until after performance starts to decrease, as can be seen in Fig. 2i for the lower of the two groups of case bases.

By analyzing the distribution of each set of initial cases (not shown here for brevity), we observe that not having enough cases in a particular area of the distribution (*i.e.*, having holes in the case base in important places) causes the case base to have difficulties to reach the best performance. We were able to confirm this effect by manually removing cases in parts of the distribution, in experiments omitted here for brevity. Conversely, if we manually make one class over-represented, the performance is not damaged as much, as the cases in the over-represented class are redundant and are removed in the plateau *(ii)*.

From these results, the initial cases harm the best performance only when the initial performance is too poor (leading to converging too slowly to reach the best state) or when there are no cases in an important area of the boundary.

Robustness w.r.t. the reference cases. The cases used to measure the competence can have a critical impact on the best performance reached. If there is no major gap between the distribution of references and the true distribution of the data, the maximal performance can be harmed but is still in the same range as the other references, as can be seen with the cyan references in Fig. 2g and 2h. However, the effect of the references becomes striking when we manually create holes in the distribution of references, in experiments omitted for brevity. In that setting, the case base becomes biased towards the incorrect distribution of the references.

5.3.4. Qualitative Analysis

Fig. 3 displays, for each dataset and for a single initialization and reference, 3 steps of the case deletion procedure: the initial case base (first column), after 10 deletion steps (second column),

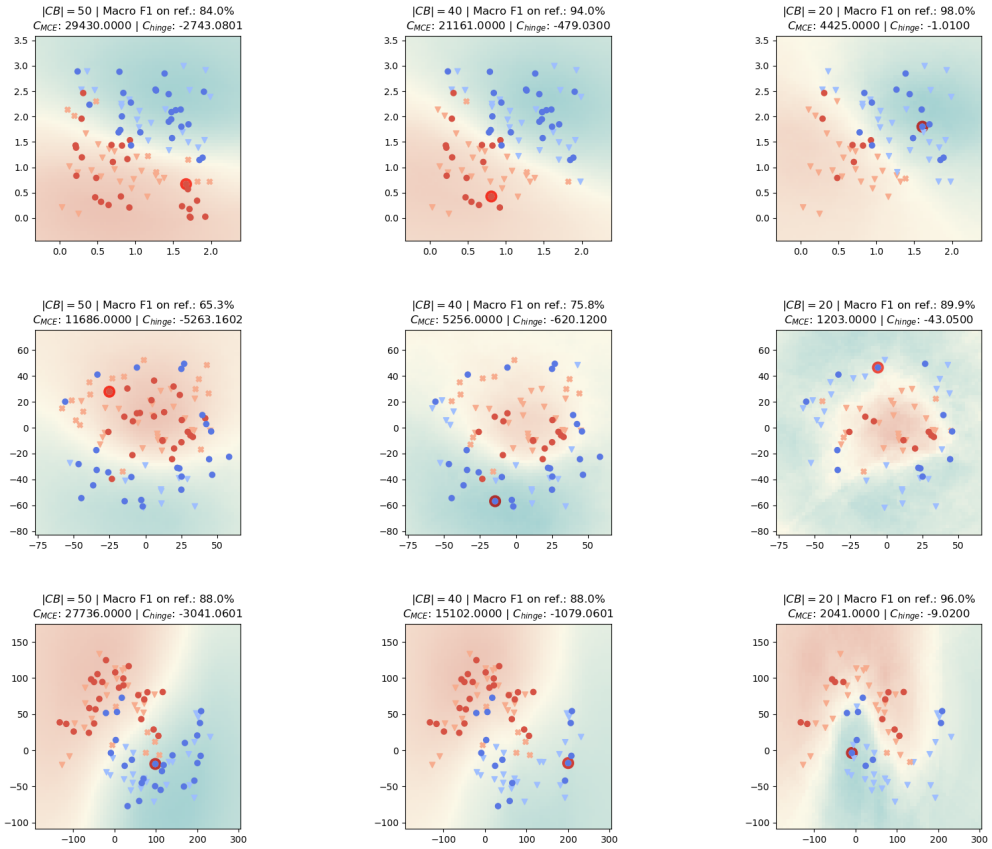


Figure 3: Three steps (first column: initial step, middle column: after 10 deletions, third column: after 30 deletions) of the case deletion procedure, for different datasets (top row: Line, middle row: Ring, bottom row: Half Moon). The case circled in red is the one that will be removed from the case base.

and after 30 deletion steps (third column). In each figure, the red and blue dots represent the remaining source cases, and the crosses and the triangles represent the references (triangles and crosses respectively mean correct and incorrect prediction). The least competent source case c_{worse} that will be deleted is circled in red.

The colored map in the figure represents CoAT's predictions for new cases across the space, with the color matching the predicted class and the saturation corresponding to the confidence (*i.e.*, the energy difference between the two outcomes, see Subsec. 4.1). In that manner, it is possible to identify the decision boundary of the compressed case base. At the end of the process, CoAT's decision frontier meets the theoretical classification boundary of the distribution, even for Half Moon, which has a relatively complex boundary. The decision frontier induced by the compressed case base is thereby able to closely approximate the ideal classification boundary.

5.4. Discussion

The compression process using C_{hinge} is able to reduce the number of cases in the case base to 40% (Ring) or even 20% (Line and Half Moon) of its initial size, while strictly improving performance. While our current experiments only cover binary classification, our approach is designed to handle any kind of nominal data in the outcome space. Further work on the approach will include multi-class classification and real-world data.

The robustness experiments show that the initial case base is not a major factor in the peak performance, as long as there are enough cases in the important regions of the situation space. However, it is important to have a proper set of reference cases, as the distribution of the references is closely matched by the compressed case base. If the reference cases are not representative of the true distribution of the data, then the compressed case base is not guaranteed to match the true distribution. To summarize, it is useful to focus on the quality of the reference (*i.e.*, how representative of the actual distribution they are) and on having sufficient initial cases for the case base, even if their quality is rather poor, as long as they cover enough of the distribution for the intended purpose of the model.

Additionally, we obtain empirical evidence of the benefits of C_{hinge} over C_{MCE}, and the performance of the case base measured by C_{hinge} correlates to CoAT's prediction performance. The question of whether this measure of competence is compatible with other CBR processes than CoAT remains open, in particular since CoAT and our competence measure are based on the same energy function. The ordering of cases—based on their competence—may change after a case is removed, as our competence measure involves the rest of the case base. This might have an effect on the compression process, but our energy-based approach to competence may offer theoretical guaranties or bounds on those changes. If the competence of a case remains stable when removing another case, we can speed up convergence by removing cases by batches.

6. Conclusion and Future Work

This paper introduced an energy-based approach to measuring the competence of a case base for machine learning tasks such as case prediction and classification. This competence approach differs from prior approaches proposed in the literature as it relies on the optimization of a global compatibility indicator between two similarity measures, one on the situation space and the other on the outcome space.

We show empirically that this notion of competence is tightly related to performance for a case-based classification task, in the sense that the competence of a source case is positively correlated to its ability to reduce the energy of correct outcomes and to increase the energy of incorrect outcomes. We analyze both quantitatively and qualitatively the behavior of this competence-based approach on different datasets (with substantially different distributions) and taking into account different classification frontiers and loss functions. Moreover, we analyze its robustness with respect to different reference and initial cases.

These results suggest the strong potential of this energy-based framework for guiding case base maintenance, providing an alternative to existing methods. One of the main differences is that it employs a global approach by considering the competence of a case base as a whole, rather than a local approach as it is often the case in the literature (where only nearest neighbors

are considered). The empirical and thorough comparison between the former and the latter will constitute one of the topics to be investigated in a future contribution.

7. Acknowledgments

This research was partially supported by the ANR project "Analogies: from theory to tools and applications" (AT2TA), ANR-22-CE23-0023 and by the ANR project "Similarity Measure Learning for Analogical Transfer" (SMeLT), ANR-22-CE23-0032. David Leake's work was funded in part by the Department of the Navy, Office of Naval Research (Award N00014-19-1-2655).

References

[1] M. M. Richter, Knowledge Containers, in: Readings in Case-Based Reasoning, 2003. URL: https://www.researchgate.net/publication/225070310_Knowledge_Containers.

[2] N. Arshadi, I. Jurisica, Maintaining case-based reasoning systems: A machine learning approach, in: P. Funk, P. A. González-Calero (Eds.), Advances in Case-Based Reasoning, 7th European Conference, ECCBR'04', volume 3155 of *Lecture Notes in Computer Science*, Springer, 2004, pp. 17–31.

[3] L. Cummins, D. Bridge, On Dataset Complexity for Case Base Maintenance, in: A. Ram, N. Wiratunga (Eds.), Case-Based Reasoning Research and Development, volume 6880, Springer, 2011, pp. 47–61.

[4] L. Cummins, Combining and Choosing Case Base Maintenance Algorithms, Ph.D. thesis, University College Cork, 2013.

[5] B. Smyth, M. Keane, Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems, in: Proc. of the 13th Int. Joint Conf. on Artificial Intelligence IJCAI, Morgan Kaufmann, 1995, pp. 377–382.

[6] B. Smyth, E. McKenna, Competence models and the maintenance problem, Computational Intelligence 17 (2001) 235–249.

[7] S. C. K. Shiu, D. S. Yeung, C. H. Sun, X. Wang, Transferring case knowledge to adaptation knowledge: An approach for case-base maintenance, Comput. Intell. 17 (2001) 295–314.

[8] J. Zhu, Q. Yang, Remembering to add: Competence-preserving case-addition policies for case base maintenance, in: Proc. of the 15th Int. Joint Conf. on Artificial Intelligence, IJCAI, Morgan Kaufmann, 1999, pp. 234–241.

[9] E. Hüllermeier, Credible case-based inference using similarity profiles, IEEE Transactions on Knowledge and Data Engineering 19 (2007) 847–858.

[10] F. Badra, A Dataset Complexity Measure for Analogical Transfer, in: Proc. of the 29th Int. Joint Conf. on Artificial Intelligence IJCAI, 2020, pp. 1601–1607.

[11] F. Badra, M.-J. Lesot, Case-Based Prediction – A Survey, IJAR 158 (2023) 108920.

[12] F. Badra, M.-J. Lesot, Theoretical and Experimental Study of a Complexity Measure for Analogical Transfer, in: ICCBR, 2022, pp. 175–189.

[13] F. Badra, M.-J. Lesot, CoAT-APC: When Analogical Proportion-based Classification Meets Case-Based Prediction, in: Proc. of the workshop on Analogies: from Theory to Applications ATA@ICCBR, CEUR-WS, 2022.

[14] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, F. J. Huang, A Tutorial on Energy-Based Learning, in: Predicting Structured Data, MIT Press, 2006.
[15] D. Wilson, D. Leake, Maintaining case-based reasoners: Dimensions and directions, Computational Intelligence 17 (2001) 196–213.
[16] G. Houeland, A. Aamodt, The utility problem for lazy learners - towards a non-eager approach, in: I. Bichindaritz, S. Montani (Eds.), Case-Based Reasoning Research and Development, ICCBR 2010, Springer, 2010, pp. 141–155.
[17] D. Leake, B. Schack, The problem drift problem and first steps towards addressing it for CBR, in: Case-Based Reasoning Research and Development, ICCBR 2023, Springer, 2023. In press.
[18] G. Hinton, S. Osindero, M. Welling, Y.-W. Teh, Unsupervised Discovery of Nonlinear Structure Using Contrastive Backpropagation, Cognitive Science 30 (2006) 725–731.

Improving Sentence Embedding With Sentence Relationships From Word Analogies

Qixuan Zhang[1,*], Yves Lepage[1,*]

[1] *Waseda University, Japan*

Abstract

In this study, we introduce a novel approach to enhance sentence embedding by leveraging word analogy. Compared with past methods that use word analogy on sentence-level tasks, our method is less affected by sentence patterns and pays more attention to semantic relations. By fine-tuning pre-trained models as BERT, RoBERTa and Sentence-BERT and evaluating their performance on inter-sentence downstream tasks, we demonstrate the efficiency of our method. Our experimental results show that each model, following fine-tuning using our approach, exhibits improvements across all inter-sentence tasks. In the STS task, our method increases the average result from 18.63% to 62.52% on BERT. This outcome substantiates that sentence relationships derived from word analogy contain valuable knowledge that can enhance the performance of sentence embedding models.

Keywords

word analogy, sentence embedding, semantic relationship

1. Introduction

Generating meaningful representations for sentences has been a subject of great interest in the field of natural language processing (NLP). Accurate sentence embeddings are crucial for a wide range of downstream tasks, including sentiment analysis and translation. Previous research, as summarized by Li et al. [1], has shown that models trained on Natural Language Inference (NLI) datasets often outperform others in various evaluation tasks. NLI datasets provide valuable world knowledge that helps sentence embedding models understand the meaning of sentences. However, creating NLI datasets, such as the Stanford Natural Language Inference (SNLI) dataset [2], requires substantial human effort, with thousands of contributors involved.

Therefore, we propose a method to generate sentence relationship data almost automatically, thus can be applied to low-resource languages with low cost. The main idea is to map the semantic relationships in the word analogy dataset to the definition sentences corresponding to the words. This process results in organized clusters of sentence relationships, which we refer to as **D**efinition **S**entences from **BATS** (DSBATS). Each DSBATS cluster contains pairs of sentences that represent specific relationships, such as the relationship between an animal and its sound (e.g., *"feline mammal usually having thick soft fur and no ability to roar"* and *"the*

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.

oakori@toki.waseda.jp (Q. Zhang); yves.lepage@waseda.jp (Y. Lepage)

0009-0006-3247-3146 (Q. Zhang); 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)

CEUR-WS.org/Vol-3438/paper_04.pdf

sound made by a cat"). In total, DSBATS based on semantic network (DSBATS-sn) consists of 20 clusters, each capturing a distinct relationship.

We employ contrastive learning and DSBATS-sn to fine-tune popular models, including BERT, RoBERTa, and Sentence-BERT. We do data augmentation on DSBATS-sn to get DSBATS for contrastive learning (DSBATS4CL). Through a series of experiments, we evaluate the performance of our approach in three inter-sentence downstream tasks. Firstly, we propose an intrinsic evaluation task called "sentence relationship similarity distinguishing", a task of identifying whether the two sentence relationships are the same. Fine-tuning the models with DSBATS4CL leads to performance improvements of 8.37%, 7.42%, and 7.87% on BERT, RoBERTa, and Sentence-BERT, respectively, compared to the performance of the original pre-trained models. Secondly, in the Semantic Textual Similarity (STS) task, our method achieves improvements of 43.6%, 21.46%, and 13.89% on the three pre-trained models, respectively. Additionally, our approach consistently produces modest improvements on the Microsoft Research Paraphrase Corpus (MRPC) dataset.

We prove that the language model can learn knowledge from sentence relationships generated from word analogy to improve the performance on semantic analysis tasks. Compared with the sentence relationships from NLI datasets, our method reduces the need for human annotation and increases the diversity of inter-sentence relations effort by using semantic network and word analogy data. Meanwhile, sentence relationship similarity distinguishing task proposed in this paper is also a challenging evaluation metric for sentence embedding method.

2. Pretrained models and sentence embedding

Pretrained models have played a significant role in the advancement of natural language processing (NLP) tasks. They are models that are pre-trained on large corpora of text data to learn language representations that capture semantic and syntactic properties of words and sentences. Transformer-based pre-trained model like BERT [3] are not only effective in word-level tasks, but also in sentence-level tasks, because of their ability to capture contextual information and because they can be simply transfered to different downstream tasks. BERT uses the [CLS] token specifically to capture a sentence-level semantics. By extracting the representation of the [CLS] token from the output, we can obtain a sentence embedding that reflects the contextual information of the entire sentence. BERT's pretraining procedure includes two specific tasks: masked language modeling (MLM) and next sentence prediction (NSP). RoBERTa [4] builds upon BERT. It introduces dynamic masking and removes next sentence prediction, leading to improved performance and robustness.

Many sentence embedding methods opt to fine-tune BERT or RoBERTa using sentence-level pre-training tasks. Sentence-BERT (SBERT) [5] is typically based on the BERT architecture. Its fine-tuning task focuses on natural language inference (NLI), aiming to train a sentence embedding space that effectively captures semantic relationships between sentences. In [1], it is noted that methods based on natural language inference (NLI) datasets exhibit excellent performance in various downstream tasks. The authors argue that the sentence relationships captured in NLI include world knowledge that can improve language models.

Table 1
Examples from BATS

Animal	Sounds
bee	buzz/hum
dog	bark/growl/howl/yelp/whine/arf/woof
cat	meow/meu/purr/caterwaul
duck	quack

3. Definition Sentences from BATS (DSBATS)

In this section, we introduce how we extract sentence relationship data, where the relationships and sentences are from, and how to connect them together. The extraction result is DSBATS based on semantic networks (DSBATS-sn). We give statistics on DSBATS-sn in Table 2 and examples in Tables 3 and 4. The dataset is available. [1]

3.1. Relationship resource: word analogy

We use word analogy as the relationship resource. The most common example of word analogy is $king : queen :: man : woman$, it states that *"king is to queen as man is to woman"*. Phenomena like this are to be studied as an important process of human cognition, with the development of language models, computational analogy has attracted more and more attention [6]. Mikolov [7] proposed to use the word offset technique to calculate this phenomenon with vectors corresponding to the words. That means, in an ideal word embedding space, the result of $\vec{king} - \vec{man} + \vec{woman}$ should be equal to $\vec{queen}$. This method is widely used as a benchmark to evaluate the quality of word embedding technique, and several word analogy test datasets have been proposed, like the Google analogy test set [7] and the Bigger Analogy Test Set (BATS) [8]. We choose BATS because it has fewer homonymy problems and various categories. BATS includes 40 morphological and semantic categories, each category can be regarded as a word analogy cluster. Example of a small word analogy cluster from BATS is shown in Table 1. Any two lines of words in the same cluster can form an analogical quadruple, like $bee : buzz :: dog : bark$.

There have also been past studies on constructing sentence relationships through word analogy, in [9]. They create general-purpose templates and replace a word that matches the word in the word analogy dataset in the template. For the sentence templates *"They traveled to **Havana**"* and *"They took a trip to **Cuba**"*, by replacing ***Havana-Cuba*** with capital-country word pairs found in word analogy datasets, a cluster of sentence pairs with similar relationships can be generated. The sentences generated by this method have the same sentence patterns. In fact, similar sentence patterns are not necessary to express similar semantics. Here by contrast, we construct sentences with semantic relationships that are not affected by sentence patterns.

[1] https://drive.google.com/drive/folders/DSBATS

Table 2
The size of categories of DSBATS-sn. The categories come from BATS. The sizes are the number of pairs of sentences.

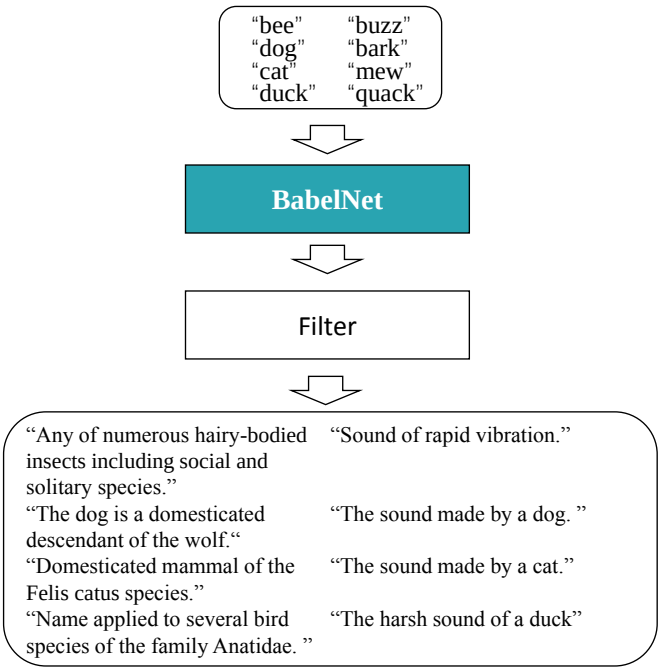
Encyclopedic	Size	Lexicographic	Size
E01 country - capital	447	L01 hypernyms - animals	4318
E02 country - language	669	L02 hypernyms - misc	5005
E03 UK city - county	426	L03 hyponyms - misc	6768
E04 name - nationality	570	L04 meronyms - substance	1312
E05 name - occupation	912	L05 meronyms -part	854
E06 animal - young	566	L06 meronyms - part	4036
E07 animal - sound	633	L07 synonyms - intensity	1645
E08 animal - shelter	877	L08 synonyms - exact	1307
E09 things - color	934	L09 antonyms - gradable	5560
E10 male - female	384	L10 antonyms - binary	1453

3.2. Sentences resource: semantic networks

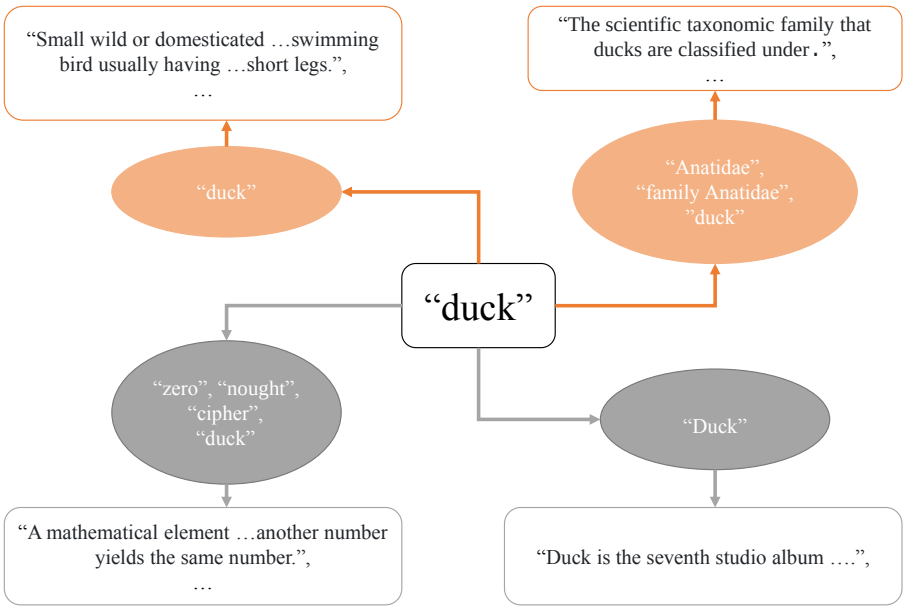
We use semantic networks as the sentences resource. Semantic network is a kind of resource in a graphical form that shows the relationships between concepts or entities. In a semantic network, concepts are represented by nodes, and the relationships between those concepts are represented by edges that connect the nodes. BabelNet is a multilingual semantic network and ontology that provides a wide range of information about words and concepts in multiple languages [10]. BabelNet integrates information from a variety of sources, including WordNet [11], Wikipedia, and other lexical and semantic resources, it currently supports over 300 languages. A synset node in BabelNet includes its synonyms set, the part of speech, the domain category, the definition sentences, and other related information. The most important information for us is the definition sentences.

3.3. Extraction process

With BATS as our relationship resource and BabelNet as our sentence resource, we build the dataset **D**efinition **S**entences from **BATS** based on sementic network (DSBATS-sn). The extraction process is as the Figure 1(a). We input word analogy clusters into BabelNet, and BabelNet will deliver several synsets for each word in the clusters. In Figure 1(b), we have "duck" and "quack" as a pair in the cluster of animal:sound relationship, the search for"duck" in BabelNet delivers 35 different synsets, including synsets that do not conform to the animal:sound relationship like the synset with number 0 in math area. We use a filter to select the valid synsets that refer to the concept corresponding to the relationship. The filter takes advantage of the information contained in BabelNet, like the domain category or the part of speech, to select the synsets that match the relationships. In Figure 1(b), orange synsets are chosen, and gray synsets are discarded. The definition sentences in the valid synsets will be organized as sentence relationship clusters as the output part in Figure 1(a). In DSBATS-sn, there are 20 clusters corresponding to 20 different relationships, the size of different clusters (categories) are shown in Table 2. Some additional examples are shown in Tables 3 and 4.



(a) Input a word analogical cluster in BATS and output a sentence relationship cluster in DSBATS-sn



(b) Filter selection

Figure 1: The process for building DSBATS-sn

Table 3
Examples extracted from word pairs in L04 category in BATS, describe the relationship of things and their substance.

Word 1	Sentence 1	Word 2	Sentence 2
atmosphere	The gases surrounding the Earth or any astronomical body.	oxygen	Chemical element.
chocolate	Chocolate is a food product made from roasted and ground cacao seed kernels, that is available as a liquid, solid or paste, on its own or as a flavoring agent in other foods.	cocoa	Good, condiment, flavor, food ingredient or product solid derived from Theobroma cacao; precursor of commercial chocolates.
cocktail	Alcoholic mixed drink	water	Chemical compound; main constituent of the fluids of most living organisms.

4. Fine-tuning with DSBATS-sn

When using DSBATS-sn for fine-tuning, we aim to have similar relationships close to each other and different relations far away from each other in the embedding space. For example, the relationship between *"Domesticated mammal of the felis catus species"* and *"The sound made by a cat"*, and the relationship between *"The dog is a domesticated descendant of the wolf"* and *"The sound made by a dog"* are both the relationship of animal:sound, they are positive examples that should be close, and we can generate sentence pairs in another relationship sound:animal as negative examples by exchange the position in the pair. This requirement conforms to the basic idea of contrastive learning, which is to narrow the distance of relevant samples and push the distance of irrelevant samples in a certain feature space. Contrastive learning does not require very large-scale labeled data, and it can make the samples more uniformly distributed in the feature space [12]. We basically follow the contrastive learning framework and configuration of [13]. Following the idea of contrastive learning, we create negative examples through the operation above and get DSBATS for contrastive learning (DSBATS4CL). One example in DSBATS4CL includes 3 relationships, that is 6 sentences, as Table 5, the red one is is the negative pair. We have 2,244,530 such examples in DSBATS4CL for training.

Our loss is basiclly InfoNCE [14], in a batch of size S, the InfoNCE loss of the ith example x_i is:

$$\text{loss}_i = -\log\left(\frac{e^{\text{sim}(x_i, {x_i}^+)/\tau}}{\sum_{j=1}^{S} e^{\text{sim}(x_i, {x_j}^+)/\tau}}\right) \tag{1}$$

But there is a little difference, we only use the third pairs in the batch as negative examples, instead of using all the samples in the same batch except for x_i as negative examples for x_i, so

Table 4
Examples extracted from word pairs in E09 category in BATS, describe the relationship of things and their color.

Word 1	Sentence 1	Word 2	Sentence 2
tomato	The tomato is the edible berry of the plant Solanum lycopersicum, commonly known as the tomato plant.	red	Red color or pigment; the chromatic color resembling the hue of blood
potato	Annual native to South America having underground stolons bearing edible starchy tubers; widely cultivated as a garden vegetable; vines are poisonous.	brown	Brown can be considered a composite color but is mainly a darker shade of red.
grass	A very large and widespread family of Monocotyledoneae, with more than 10.000 species, most of which are herbaceous, but a few are woody. The stems are jointed, the long, narrow leaves originating at the nodes. The flowers are inconspicuous, with a much reduced perianth, and are wind-pollinated or cleistogamous.	green	A colour sometimes referred to as Luggage or Luggage Green

Table 5
DSBATS4CL example

Sentences	Relationship
Any of numerous hairy-bodied insects including social and solitary species. Sound of rapid vibration.	animal:sound
The dog is a domesticated descendant of the wolf. The sound made by a dog.	animal:sound
Sound of rapid vibration. Any of numerous hairy-bodied insects including social and solitary species.	sound:animal

our loss of x_i is:

$$\text{loss}_i = -\log\left(\frac{e^{\text{sim}(x_i, x_i^+)/\tau}}{\sum_{j=1}^{S} e^{\text{sim}(x_i, x_j^-)/\tau}}\right) \tag{2}$$

x_j^- corresponding to the red example in Table 5.

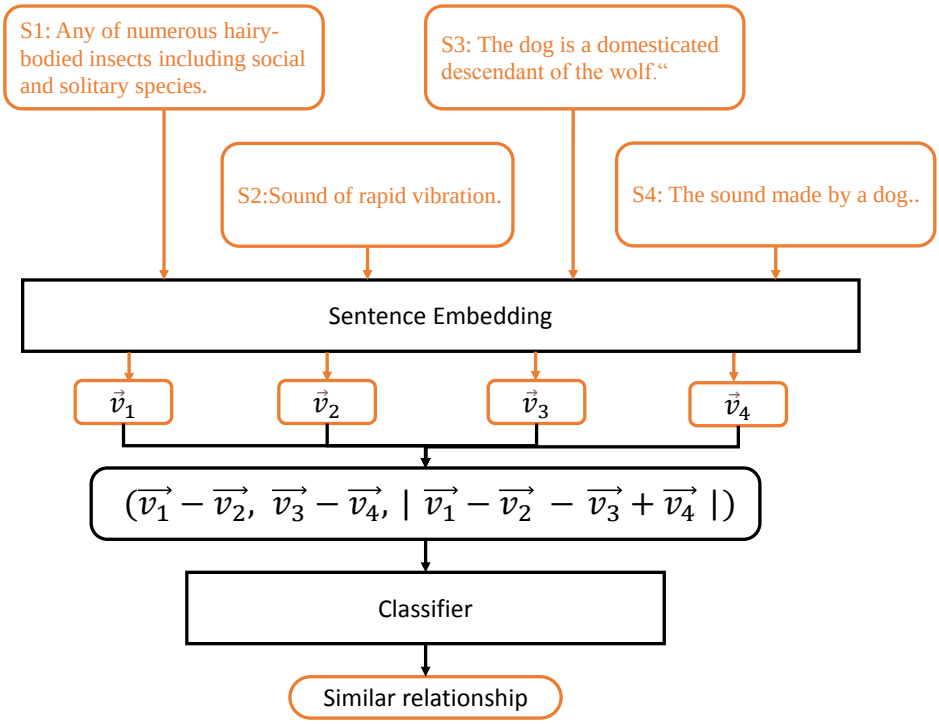


Figure 2: Sentence relationship similarity distinguishing (SRSD) task

5. Experiment and evaluation

5.1. Intrinsic evaluation

We designed the **S**entence **R**elationship **S**imilarity **D**istinguishing (SRSD) task as intrinsic evaluation. It inputs a pair of relationships at a time, which is 4 sentences, and predicts if they are two similar relationships. Figure 2 shows the process with two sentence relationships in the same category. The test set is a manually annotated DSBATS dataset different from the automatically extracted DSBATS-sn in Section 3. The manual version uses the same relationship resource BATS but different sentence resources like Oxford Dictionary, Webster's Dictionary, and Collins Dictionary, so we call the manually annotated DSBATS as DSBATS-dic. The number of pairs of sentences in DSBATS-dic is shown in Table 6. After learning with DSBATS4CL, all three models improved accuracy on this task. The improvement is basically about 6%. The best performance is from Sentence-Bert with DSBATS4CL, which reaches 69.55%. Table 7 shows the result of SRSD task.

Table 6
Statistics on DSBATS-dic

Category	Size
L01 hypernyms - animals	251
L02 hypernyms - misc	225
L04 meronyms - substance	127

Table 7
Evaluation result of all the tasks.

		Intrinsic eval.	Extrinsic eval.	
Model	DSBATS4CL	SRSD	STS avg.	MRPC
BERT	w/o	58.18	18.63	68.81
	w/	**64.27**	**62.53**	**70.14**
RoBERTa	w/o	58.47	43.65	71.42
	w/	**65.83**	**65.11**	**71.83**
SBERT	w/o	61.68	62.84	73.51
	w/	**69.55**	**77.56**	**74.20**

5.2. Extrinsic evaluation

We conducted extrinsic evaluations using the Semantic Textual Similarity (STS) and Microsoft Research Paraphrase Corpus (MRPC) datasets as extrinsic evaluations. We use SentEval [15] to do the evaluation and follow the default configurations. The STS evaluation involves inputting two sentences and predicting a score between 0 and 5 that represents the similarity between the two sentences. Higher scores indicate better performance, as they align more closely with human-labeled similarity. The results, as summarized in Tables 7 and 8. They demonstrate the impact of fine-tuning with DSBATS4CL on the performance of the three pre-trained models. After fine-tuning with DSBATS4CL, the performance of all three pre-trained models improves. Notably, BERT and RoBERTa, which had not previously learned the relationship between sentences, improve by 43.89% and 20.02% in average, respectively. MRPC input two sentences and predict if they are similar or not. Higher scores correspond to higher accuracy. In comparison to the STS task, the improvements on MRPC are relatively small. The best performance is achieved by Sentence-BERT with DSBATS4CL, attaining accuracy of 66.06% on STS and 74.20% on MRPC, respectively. The results indicate that knowledge captured from sentence relationships derived from word analogy is valuable, fine-tuning with DSBATS4CL enhances the models' ability to understand the semantic relation between sentences.

6. Conclusion

In this work, we introduced a method to enhance sentence embedding using word analogy. We map the relationships between words to relationships between sentences by using definition

Table 8
Result on STS. STSB stands for STSBenchmark, SICK-R stands for SICK Relatedness. The last column is the same as in Table 7.

Model	DSBATS4CL	STS12	STS13	STS14	STS15	STS16	STSB	SICK-R	STS avg.
BERT	w/o	7.19	29.06	12.55	16.16	28.92	6.43	30.11	18.63
	w/	**54.29**	**68.37**	**58.54**	**67.23**	**69.21**	**60.80**	**59.25**	**62.53**
RoBERTa	w/o	16.73	45.56	30.24	55.27	56.87	39.14	61.76	43.65
	w/	**53.22**	**67.60**	**60.96**	**69.50**	**71.17**	**68.52**	**64.81**	**65.11**
SBERT	w/o	64.92	65.56	65.79	63.66	60.92	62.49	56.51	62.84
	w/	**71.85**	**82.21**	**79.85**	**82.44**	**77.67**	**77.54**	**71.35**	**77.56**

sentences in semantic network. Compared with the past methods that use word analogy in sentence-level tasks by replacing words in sentences, our method is less limited by morphology and pays more attention to semantics. The improvements on downstream tasks like STS and MRPC prove that the sentence relationships from word analogy include the knowledge that can enhance the semantic understanding of sentence embedding models. Sentence relationship similarity distinguishing task proposed as an intrinsic evaluation in our work can also be a challenging evaluation task for other sentence embedding methods. We believe that it is worth further exploring ways to combine analogy with contrastive learning, as analogy relation has many equivalent forms suitable for contrastive learning to construct positive and negative examples.

Acknowledgments

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

References

[1] R. Li, X. Zhao, M.-F. Moens, A brief overview of universal sentence representation methods: A linguistic view, ACM Computing Surveys (CSUR) 55 (2022) 1–42.

[2] 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, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 632–642. URL: https://aclanthology.org/D15-1075. doi:10.18653/v1/D15-1075.

[3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational

Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[4] L. Zhuang, L. Wayne, S. Ya, Z. Jun, A robustly optimized BERT pre-training approach with post-training (2021) 1218–1227. URL: https://aclanthology.org/2021.ccl-1.108.

[5] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3982–3992.

[6] H. Prade, G. Richard, Computational Approaches to Analogical Reasoning: Current Trends, volume 548, Springer Publishing Company, Incorporated, 2014.

[7] T. Mikolov, K. Chen, G. S. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: International Conference on Learning Representations, 2013.

[8] A. Gladkova, A. Drozd, S. Matsuoka, Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't., in: Proceedings of the NAACL Student Research Workshop, Association for Computational Linguistics, San Diego, California, 2016, pp. 8–15. URL: https://aclanthology.org/N16-2002. doi:10.18653/v1/N16-2002.

[9] X. Zhu, G. de Melo, Sentence analogies: Linguistic regularities in sentence embeddings, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 3389–3400.

[10] R. Navigli, S. P. Ponzetto, Babelnet: Building a very large multilingual semantic network, in: Proceedings of the 48th annual meeting of the association for computational linguistics, 2010, pp. 216–225.

[11] G. A. Miller, Wordnet: a lexical database for english, Communications of the ACM 38 (1995) 39–41.

[12] T. Wang, P. Isola, Understanding contrastive representation learning through alignment and uniformity on the hypersphere, in: International Conference on Machine Learning, PMLR, 2020, pp. 9929–9939.

[13] T. Gao, X. Yao, D. Chen, SimCSE: Simple contrastive learning of sentence embeddings, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 6894–6910. URL: https://aclanthology.org/2021.emnlp-main.552. doi:10.18653/v1/2021.emnlp-main.552.

[14] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, CoRR abs/1807.03748 (2018). URL: http://arxiv.org/abs/1807.03748. arXiv:1807.03748.

[15] A. Conneau, D. Kiela, SentEval: An evaluation toolkit for universal sentence representations, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018. URL: https://aclanthology.org/L18-1269.