Transformer25 at SemEval-2025 Task 1: A similarity-based approach

Wiebke Petersen, Lara Eulenpesch, Ann Maria Piho, Julio Julio, Victoria Lohner
Heinrich Heine Universität
Düsseldorf, Germany
{petersew, laeul100, anpih100, jujul100, pic28faw}@hhu.de

Abstract

Accurately representing non-compositional language, such as idiomatic expressions, is crucial to prevent misinterpretations that may affect subsequent tasks. This paper presents our submission to the SemEval 2025 task on advancing the representation of multimodal idiomaticity. The challenge involves matching idiomatic expressions with corresponding image descriptions that depict their meanings. We participate in the text-only tracks of both subtasks. Our system adopts a similarity-based approach and utilizes embeddings from pre-trained BERT-based large language models alongside ChatGPT-generated textual content. The primary goal is to explore the extent to which semantic similarity of embeddings from pre-trained models can effectively represent idiomaticity. For subtask A, our final submission ranked 5th on the test data and 3rd on the extended evaluation data (both out of 6).

1 Introduction

Idiomaticity of multiword expressions (MWEs) – the gap between the literal meaning of individual parts and the figurative meaning of the whole – is a major source of the lexical, syntactic and semantic quirks that make MWEs notoriously challenging for NLP systems (Baldwin and Kim, 2010). It is no surprise, that (Sag et al., 2002) dubbed MWEs 'a pain in the neck.'

Expressions like 'black sheep' have a literal and an idiomatic meaning. Identifying the intended meaning in context is crucial for tasks such as machine translation and question answering. While humans can easily distinguish between idiomatic and literal usage, language models often struggle. Addressing this challenge is the focus of the SemEval 2025 task *AdMIRe: Advancing Multimodal Idiomaticity Representation* (Pickard et al., 2025), an extension of the 2022 task *Multilingual Idiomaticity Detection and Sentence Embedding* (Tayyar Madabushi et al., 2022). Both tasks focus on nominal compounds. While the earlier task focused on classifying idiomatic versus literal uses of nominal compounds in context, the new task introduces a multimodal component, requiring models to select appropriate images (or image captions) based on the intended meaning. Two subtasks are defined: In Subtask A, images have to be ranked based on how closely they relate to a noun compound used idiomatically or literally in a given sentence. In Subtask B, the task is to decide which image best completes a given 2-element image sequence and to decide whether the image sequence illustrates the idiomatic or the literal meaning of the compound in question.

In this paper we address Subtasks A and B in their monolingual, English, text-only version – using image captions rather than images themselves. Our approach relies on comparing the similarity of contextualized compound and sentence embeddings. The central question we explore is whether these tasks can be effectively tackled without specialized training or fine-tuning, using only contextualized embeddings from pre-trained large language models. In line with the famous essay title by Vaswani et al. (2017), we ask, "Is similarity all you need?"[1]

2 Background

For **Subtask A**, the given data consists of a nominal compound, a sentence in which it is used either literally or idiomatically, and five images accompanied by detailed textual descriptions (referred to as captions). These images vary in how closely they relate to the possible meanings of the compound: one represents the idiomatic and one the literal meaning, two are semantically related (one to the idiomatic and one to the literal meaning), and one functions as a distractor. The distractor image is not directly related to the compound but

[1] The code of our approach can be found at https://github.com/WiebkePetersen/Transformer25.

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 2311–2317
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

may come from a similar semantic domain. For instance, in the case of the compound 'rotten apple,' a distractor might be a sugar-coated peach.

The task is to rank the images as follows: (i) If the compound is used literally in the sentence, the desired order is literal, related-to-literal, related-to-idiomatic, idiomatic, distractor. (ii) If the compound is used idiomatically, this order is reversed, except that the distractor remains in the final position: idiomatic, related-to-idiomatic, related-to-literal, literal, distractor. For evaluation the system's predicted rankings are compared to the expected ones using two metrics. The first is Top Image Accuracy (or top-1 accuracy), which checks whether the system correctly identifies the most representative image (literal or idiomatic, depending on usage) by ranking it first. The second is Rank Correlation, which assesses the overall alignment of the predicted ranking with the gold standard using Spearman's rank correlation coefficient.

For **Subtask B** only the nominal compound is provided but no sentence containing it. Instead, a sequence of two images with captions and four additional candidate images with captions are given. The task is twofold: (i) determine whether the initial image sequence represents an idiomatic or literal use of the compound; (ii) out of the four additional images choose the one that best completes the sequence. The four images are composed such that one is the optimal continuation for the idiomatic interpretation and one for the literal interpretation of the compound. The remaining two images are semantically related to the first two but are not ideal completions, analogous to the related images in Subtask A. Evaluation of the subtask is based on the accuracy of both predictions: (i) identifying the correct type (literal vs. idiomatic) and (ii) selecting the appropriate image to continue the sequence.

The English datasets used in the tasks are based on the data from the 2022 task and comprise 250 nominal compounds. The images for both subtasks were generated using Midjourney v6.0, based on prompts created with Gemini Pro 1.5 to capture the relevant meaning nuances. Context sentences for the compounds were either sourced from the web or written specifically for this task. The data is divided into training, development, test, and extended evaluation sets. Table 1 provides an overview of the dataset sizes and the distribution of idiomatic and literal instances across these splits.

Subtask A data set	**# sentences**	**idiomatic / literal**
Training	70	39 / 31
Development	15	7 / 8
Test	15	8 / 7
Extended Evaluation/Test	100	46 / 54

Subtask B data set	**# compounds**	**idiomatic / literal**
Training	20	13 / 7
Development	5	2 / 3
Test	5	3 / 2
Extended Evaluation/Test	30	12 / 18

Table 1: Summary of datasets for Subtask A and B.

3 System overview

For both subtasks, we participate in the monolingual, English, text-only track, which means that we rely solely on image captions and do not use the images themselves. Our approach is based on computing similarity scores between embeddings of the provided textual material (sentences and captions) as well as additional texts that we generate automatically. Subsection 3.1 outlines the process of generating this additional material and computing embeddings. The following subsections, 3.2 and 3.3, describe how predictions for Subtasks A and B are derived from the computed similarity scores.

3.1 Data Extension and Embeddings

We augment the training data using the prompt-based strategy proposed by Dai et al. (2025), who demonstrate that synthetic samples generated with ChatGPT can improve performance in low-resource settings. For each compound we generate additional textual data using ChatGPT-4.[2] The model is prompted to generate for each compound (i) one definition each for its literal and idiomatic meanings, ensuring that the compound occurs in the definition; (ii) two example sentences using the compound, again one with its literal and one with its idiomatic meaning; (iii) for each meaning (literal and idiomatic) a caption for an image illustrating the compound. Definitions are included as Tsukagoshi et al. (2021) show that definition sentences may improve semantic textual similarity tasks. The prompts used, together with an example of generated definitions, sentences, and image captions are shown in Table 2.

Embeddings for both the GPT-generated and the provided textual data are extracted using two pre-trained language models: the standard BERT-model `bert-base-uncased` (Devlin et al., 2019,

[2] `https://chatgpt.com/` (accessed in November 2024)

prompt: definitions and sentences (literal and idiomatic)	*I will give you expressions that have both a literal and an idiomatic meaning. Define each meaning, starting with “... is.” Additionally, provide an example sentence using the expression.*
prompt: image captions	*I will give you expressions (mainly compounds) with both literal and idiomatic meanings. Provide descriptions of images illustrating both meanings. Start with “The image depicts ...”.*

Output for ‘piece of cake’

idiomatic definition	‘Piece of cake’ is a metaphor for something very easy to accomplish.
idiomatic sentence	The math test was a piece of cake; she finished in ten minutes.
idiomatic image caption	The image depicts a student effortlessly solving a problem or task, symbolizing something very easy to accomplish.
literal definition	‘Piece of cake’ is a literal term referring to a portion of a cake.
literal sentence	He cut a small piece of cake to enjoy with his coffee.
literal image caption	The image depicts a literal slice of cake on a plate, emphasizing the literal meaning of a ‘piece of cake.’

Table 2: ChatGPT-4 prompts used for data extension, with output exemplified for the nominal compound ‘piece of cake.’

in this paper referred to as BERT)[3] and a BERT-based sentence transformer `all-MiniLM-L6-v2` (Reimers and Gurevych, 2019, referred to as SBERT).[4]. The sentence embeddings from pre-trained BERT models are known to capture semantic meaning of sentences poorly (Li et al., 2020), which is why we include a specific sentence transformer that is pre-trained to perform well in semantic sentence comparison tasks.

For the BERT-embeddings, we experiment with three (pooling) methods: (a) the standard CLS-token embedding from the last hidden layer, as used for next sentence prediction during pre-training; (b) sequence mean pooling, where the element-wise arithmetic mean of the token-level embeddings from the last n hidden layers is computed; and (c) contextualized compound embeddings for texts that consistently contain the target compound (i.e., original and GPT-generated sentences and definitions). These contextualized compound embeddings are obtained by averaging the token embeddings corresponding to the compound itself across the last n hidden layers. With methods (b) and (c), we aim to better preserve the semantic information specific to the compound compared to the standard CLS embedding (a).

3.2 Subtask A

A series of experiments are conducted in order to develop the final system.

3.2.1 Experiment 1 (using only given data)

In the first experiment, our aim is to investigate how far similarity-based approaches can take us when using only the provided data. For each compound, we compute embeddings of the given context sentence and the five image captions using the models and pooling strategies described in Section 3.1. The images are ranked according to the cosine similarity of their embeddings to the sentence embedding.

Additionally, we explore whether preprocessing the data to reduce noise can improve modeling effectiveness. Our preprocessing pipeline includes text normalization by removing capitalization, special characters, extra whitespace, and punctuation. Moreover, we lemmatize all words and retain only nouns, adjectives and verbs.

For the image captions, we also experiment with shortening them to remove information that is not relevant to the content, such as the style or background of the image. However, since this does not lead to consistent improvements, we do not pursue this further.

Discussion of results: Results of Experiment 1 on the training data are presented in Table 3 for both unaltered and preprocessed data. A major observation is the overall weak rank correlations and the strong imbalance between literal and idiomatic expressions when analyzed separately.

Overall, the results for idiomatic compounds are consistently poor across all settings. The highest top-1 accuracy for idiomatic compounds without preprocessing (0.28) is achieved using the BERT model with the `meanLast` pooling strategy, which computes the mean of all token embeddings from the last hidden layer. This model configuration shares the best overall top-1 accuracy across all data with the `SBERT` model. SBERT, however, performs best on literal compound uses, showing a

[3] `https://huggingface.co/google-bert/bert-base-uncased`

[4] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

without preprocessing:			
method	**all data**	**idiomatic**	**literal**
SBERT	**0.40** (0.20)	0.18 (0.12)	**0.68** (0.31)
BERT CLS	0.21 (-0.01)	0.15 (-0.12)	0.29 (0.12)
BERT sequence mean pooling			
mean2ndToLast	0.37 (0.07)	0.18 (-0.03)	0.61 (0.20)
meanLast4	0.36 (0.10)	0.21 (0.02)	0.55 (0.19)
meanLast	**0.40** (0.01)	**0.28** (-0.10)	0.55 (0.15)
BERT contextualized compound			
mean2ndToLast	0.26 (0.09)	0.05 (0.02)	0.52 (0.18)
meanLast4	0.23 (0.07)	0.03 (-0.06)	0.48 (0.24)
meanLast	0.26 (0.03)	0.08 (-0.11)	0.48 (0.21)
with preprocessing:			
method	**all data**	**idiomatic**	**literal**
SBERT	**0.46** (0.21)	0.21 (0.13)	**0.77** (0.30)
BERT CLS	0.27 (0.05)	0.18 (0.02)	0.39 (0.10)
BERT sequence mean pooling			
mean2ndToLast	0.33 (0.08)	0.21 (0.10)	0.48 (0.05)
meanLast4	0.31 (0.09)	0.21 (0.11)	0.45 (0.08)
meanLast	0.36 (0.10)	**0.26** (0.05)	0.48 (0.15)
BERT contextualized compound			
mean2ndToLast	0.31 (0.17)	0.10 (0.09)	0.58 (0.27)
meanLast4	0.31 (0.11)	0.10 (0.04)	0.58 (0.21)
meanLast	0.29 (0.09)	0.13 (-0.05)	0.48 (0.25)

Table 3: (Experiment 1) Top-1 accuracy (rank correlation) for ranking images by cosine similarity to the original sentence, using different models and various pooling methods (mean2ndToLast: average over token embeddings of the 2nd last hidden layer, meanLast4: average over token embeddings of the 4 last hidden layers, meanLast: average over token embeddings of the last hidden layer; see Section 3.1 for details).

clear advantage for more compositional meanings.

Another interesting finding is that the sentence transformer SBERT clearly outperforms the standard CLS embeddings from BERT (0.40 vs. 0.21 top-1 accuracy), highlighting that using a model specialized in capturing sentence-level semantics is beneficial for the task.

Preprocessing negatively affects the BERT mean sequence pooling results but improves performance for both SBERT and BERT CLS embeddings, which is surprising since our radical preprocessing method removes much of the sentence structure. In addition, the contextualized compound models also benefit from preprocessing.

Overall, the best results are achieved using the SBERT model with preprocessing, which reaches a top-1 accuracy of 0.46. This configuration also yields the highest rank correlation (0.21).

3.2.2 Experiment 2 (using GPT-data)

The core idea of the second experiment is to use the GPT-generated data described in Section 3.1 alongside the gold label information about idiomaticity provided in the training data. In this setting, the image captions are ranked based on their cosine similarity to the GPT-generated comparators (definition, sentence, caption), which are selected according to the given idiomatic or literal label. For this experiment, we focus on the two best-performing model configurations from Experiment 1: SBERT and BERT with meanLast pooling.

It is important to note that the prompts used for generating definitions enforce similar phrasing at the beginning of each definition. To evaluate whether this phrasing bias affects the results, we also include a *cut* version of the definitions, where standardized introductory phrases are removed before obtaining the embeddings. Specifically, for idiomatic uses, we remove the phrase "[...] is a metaphor for", and for literal uses, we remove "[...] literal".

without preprocessing		
comparator	**SBERT**	**BERT meanLast**
GPT-caption	**0.61** (0.13)	0.49 (0.14)
GPT-definition	0.37 (0.19)	0.37 (0.05)
GPT-definition cut	**0.61** (0.16)	0.47 (0.15)
GPT-sentence	0.36 (0.06)	0.29 (0.09)
with preprocessing		
comparator	**SBERT**	**BERT meanLast**
GPT-caption	**0.61** (0.16)	0.49 (0.14)
GPT-definition	0.44 (0.29)	0.47 (0.17)
GPT-definition cut	0.53 (0.19)	0.51 (0.13)
GPT-sentence	0.39 (0.15)	0.36 (0.14)

Table 4: (Experiment 2) Top-1 accuracy (rank correlation) for ranking image captions based on similarity to GPT-generated texts: captions, (cut) definitions, sentences.

Discussion of results: Results of Experiment 2 on the training data can be found in Table 4. Compared to Experiment 1, no stronger correlations can be observed. However, the top-1 accuracy has improved significantly, which is expected as the idiomatic/literal information is now taken into account. Accordingly, experiment 3 aims to automatically assign the idiomatic/literal label. Preprocessing does not provide a clear picture, showing both improvements and degradations in performance.

As expected, the GPT-caption is most suitable as a comparator, as it is compared against image captions. Notably, for SBERT, the cut GPT-definition performs just as well. This suggests that SBERT benefits particularly strongly when repetitive elements are removed from the sentences.

3.2.3 Experiment 3 (idiomaticity classifier)

In order to make use of the GPT-generated additional texts, it is necessary to classify the sentence as idiomatic or literal. Our classifier relies solely on cosine similarity of the sentence to a comparator: the sentence is either compared to the two GPT-sentences (literal and idiomatic), to the two GPT-definitions, or to the two GPT-captions. In each case, the higher similarity determines the class label. For the embeddings, we compare the methods described in Section 3.1.

As prior work (e.g. Taslimipoor et al., 2020) shows that even a modest amount of task-specific fine-tuning can noticeably improve MWE performance on unseen data, for comparison we additionally fine-tune the BERT-model on idiomatic/literal classification using HuggingFace's trainer[5] and training for 7 epochs and use the same similarity-based classification strategy as before.[6]

comparator	pooling method	accuracy
Sentence embeddings		
GPT-sentence	SBERT	0.79
GPT-sentence	BERT CLS	0.83
GPT-sentence	BERT meanLast	0.86
GPT-definition	SBERT	0.81
GPT-definition	BERT CLS	0.66
GPT-definition	BERT meanLast	0.76
GPT-definition cut	SBERT	0.57
GPT-definition cut	BERT CLS	0.59
GPT-definition cut	BERT meanLast	0.76
GPT-caption	SBERT	0.60
GPT-caption	BERT CLS	0.61
GPT-caption	BERT meanLast	0.79
pre-trained BERT compound embedding		
GPT-sentence	BERT meanLast	0.857
GPT-sentence	BERT meanLast4	**0.9**
GPT-definition	BERT meanLast	0.671
GPT-definition	BERT meanLast4	0.743
fine-tuned BERT compound embedding		
GPT-sentence	BERT meanLast4	**0.97**

Table 5: (Experiment 3) Accuracy of idiomaticity classifier: sentence embedding is compared to comparator embedding using the specified pooling method (see Section 3.1).

Discussion of results: Table 5 shows the accuracy scores. We report results for the best mean sequence pooling method (BERT meanLast), the two best compound pooling strategies (meanLast and meanLast4), and the sentence embedding models SBERT and BERT CLS. The results show that the generated sentences perform better as comparators than the generated definitions. The cut definitions that performed very well in experiment 2 perform worse than the intact ones in experiment 3. Furthermore, compound-based embeddings outperform sentence-based ones. Pooling over the last four hidden layers also improves accuracy compared to pooling over the last layer only.

It turns out, that comparing compound embeddings obtained from the fine-tuned model using the meanLast4 pooling strategy of the GPT generated sentences and the given sentences results in the most accurate idiomaticity classifier (0.97).

3.2.4 Experiment 4 (ranking improvement)

The final experiment builds on the classifier from Experiment 3 to improve the ranking. In Experiments 1 and 2, all five captions were ranked by their similarity to a single comparator, resulting in poor correlation scores. This setup serves as our *baseline* ranker. For classification, we use the best-performing setting from Experiment 3 (see Table 5): comparing meanLast4 compound embeddings of the given and GPT-generated sentences using the pre-trained BERT model, as we aim at investigating how far we can get without fine-tunig. For ranking, we use the GPT-captions as the comparator and SBERT as the embedding model, which together performed best in Experiment 2 (see Table 4). We propose two improved ranking algorithms:

The *pair ranker* first selects the caption most similar to the literal GPT-caption ('literal1') and the one most similar to the idiomatic GPT-caption ('idiomatic1'). Next, it identifies 'literal2' and 'idiomatic2' as the captions most similar to 'literal1' and 'idiomatic1', respectively, among the remaining captions. The leftover caption is marked as 'unrelated'. If the classifier labels the sentence as literal, the predicted order is: literal1-literal2-idiomatic2-idiomatic1-unrelated; otherwise, it is: idiomatic1-idiomatic2-literal2-literal1-unrelated.

The *extreme ranker* differs from the pair ranker only in how 'literal2' and 'idiomatic2' are chosen. They are the captions (out of the remaining captions) with the second highest similarity to the according (literal/idiomatic) GPT-caption.

Results (see Table 6) show that both improved rankers outperform the baseline clearly, with little difference between the two new methods.

[5] https://huggingface.co/

[6] Fine-tuned model: https://huggingface.co/jlsalim/bert-uncased-idiomatic-literal-recognizer

method	train	test	xe
baseline (experiment 1)	0.201	0.053	0.169
with GPT-data and classifier			
baseline (experiment 2)	0.086	0.027	0.14
pair ranker	**0.324**	**0.18**	0.298
extreme ranker	0.246	0.087	**0.334**
top-1 accuracy	0.56	0.47	0.48

Table 6: Experiment 4: Rank correlations for various ranking methods on different datasets (xe: extended evaluation) and top-1 accuracy (computed with cosine similarity).

3.3 Subtask B

The same embedding strategies are used for Subtask B as for Subtask A. For classifying the image sequence as idiomatic or literal, the captions are compared to the GPT-generated data (see Section 3.1. The best results are achieved by comparing both image captions of the series to all GPT-generated data (GPT-sentence, GPT-definition, GPT-caption, each in idiomatic and in literal version). The pairing of GPT-generated data and image caption with the highest similarity predicts the label for the sequence. This method outperforms alternative methods such as averaging over similarities per class or per GPT-text-type.

Our basic approach to the second part, selecting the matching final image caption to continue a sequence of two image captions, is to determine the best fit based on similarities between SBERT embeddings in line with the approach for Subtask A. For this, the embeddings for each of the four potential final captions are compared to the average of the two previous captions of the sequence by calculating the cosine similarity. The caption scoring the highest similarity is chosen as the matching one to complete the sequence.

4 Results

Our best performing system for Subtask A uses the fine-tuned BERT-model and a compound-based mean pooling over the last four hidden states to predict whether the compound is used literally or idiomatically in the sentence. Based on this classification the pair-based ranking method predicts the ranking of the five image captions. No preprocessing is applied.

On the final test data the system reaches a top-1 accuracy of 0.47 and on the extended evaluation dataset an accuracy of 0.54. The correlation score for the test set is 2.82 and for the extended evaluation dataset 3.04. With these results the system ranks 5 out of 6 participating teams on the test set and 3 out of 6 on the extended evaluation set.

For Subtask B only two teams competed and our system described in Section 3.3 came out second on the test set (image selection accuracy: 0.8, sentence type prediction: 0.6) and first on the extended evaluation set (image selection accuracy: 0.6, sentence type prediction: 0.9).

5 Limitations and conclusion

Our approach relies heavily on GPT-generated data. While effective, better prompt design or more careful postprocessing – such as trimming irrelevant parts of captions (e.g. describing image backgrounds) – could further improve results. We have experimented with automatically cutting off caption endings but without consistent gains.

Throughout all experiments, we consistently have used cosine similarity to compare embeddings. Exploring alternative distance metrics could be a promising direction for future work (thanks to our reviewer for the suggestion). In an initial test, we apply negative Manhattan distance and observe a significant improvement for the pair ranker from Experiment 4, while other rankers show no consistent gains (see Table 7).

method	train	test	xe
baseline (experiment 1)	0.166	0.107	0.079
with GPT-data and classifier			
baseline (experiment 2)	0.111	0.133	0.119
pair ranker	**0.376**	**0.313**	**0.367**
extreme ranker	0.284	0.2	0.331
top-1 accuracy	0.54	0.47	0.51

Table 7: Same data as in Table 6 but computed with negative Manhattan distance

The experiments demonstrate that even without fine-tuning, the raw embeddings from BERT and SBERT models contain enough semantic information to solve the task via similarity comparisons. Moreover, using GPT-generated examples for idiomatic and literal uses significantly boosts performance, highlighting the value of synthetic data in semantic modeling. Overall, our findings suggest that simple similarity-based methods, supported by carefully generated auxiliary data, offer a strong baseline for idiomaticity detection and related ranking tasks.

6 Acknowledgement

The work discussed in this paper is a collaborative effort by a seminar group. We would like to thank the students of the bachelor seminar ‘Transformers in CL’ at the Heinrich Heine University Düsseldorf in the winter term 2024/2025 for helpful hints and questions.

Furthermore, we thank the anonymous reviewers for their valuable comments and suggestions, and the organizers of the shared task for their tremendous effort and dedication in setting up and running the competition.

References

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. In Fred J. Damerau Nitin Indurkhya, editor, *Handbook of Natural Language Processing*, 2 edition, pages 267–292.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Fang Zeng, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2025. AugGPT: Leveraging ChatGPT for text data augmentation. *IEEE Transactions on Big Data*, pages 1–12.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Thomas Pickard, Aline Villavicencio, Maggie Mi, Wei He, Dylan Phelps, Carolina Scarton, and Marco Idiart. 2025. SemEval-2025 Task 1: AdMIRe - Advancing Multimodal Idiomaticity Representation. In *Proceedings of the 19th International Workshop on Semantic Evaluations (SemEval-2025)*, Vienna, Austria. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. 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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Computational Linguistics and Intelligent Text Processing*, pages 1–15, Berlin, Heidelberg. Springer Berlin Heidelberg.

Shiva Taslimipoor, Sara Bahaadini, and Ekaterina Kochmar. 2020. MTLB-STRUCT @parseme 2020: Capturing unseen multiword expressions using multi-task learning and pre-trained masked language models. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 142–148, online. Association for Computational Linguistics.

Harish Tayyar Madabushi, Edward Gow-Smith, Marcos Garcia, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2022. SemEval-2022 task 2: Multilingual idiomaticity detection and sentence embedding. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 107–121, Seattle, United States. Association for Computational Linguistics.

Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. 2021. DefSent: Sentence embeddings using definition sentences. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 411–418, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.