Unsupervised Paraphrasing of Multiword Expressions

Takashi Wada[1,3]* **Yuji Matsumoto**[2] **Timothy Baldwin**[1,3] **Jey Han Lau**[1]
[1] School of Computing and Information Systems, The University of Melbourne
[2] RIKEN Center for Advanced Intelligence Project (AIP)
[3] Department of Natural Language Processing, MBZUAI
twada@student.unimelb.edu.au, tb@ldwin.net
yuji.matsumoto@riken.jp, jeyhan.lau@gmail.com

Abstract

We propose an unsupervised approach to paraphrasing multiword expressions (MWEs) in context. Our model employs only monolingual corpus data and pre-trained language models (without fine-tuning), and does not make use of any external resources such as dictionaries. We evaluate our method on the SemEval 2022 idiomatic semantic text similarity task, and show that it outperforms all unsupervised systems and rivals supervised systems.[1]

1 Introduction

Multiword expressions (MWEs) are notoriously difficult to model because the meaning of the whole can diverge substantially from that of the component words, e.g. the meaning of *swan song* ("final performance") is far removed from its component words[2] (Sag et al., 2002; Baldwin and Kim, 2010). This hampers the capacity of pre-trained language models such as BERT (Devlin et al., 2019) to capture MWE semantics (Tayyar Madabushi et al., 2021; Zeng and Bhat, 2022). Similarly, non-native speakers tend to have difficulty understanding MWEs, especially those that have no equivalent in their native language (Irujo, 1986; Arnaud and Savignon, 1996).

In this work, we propose a method for paraphrasing non-literal MWEs (e.g. *swan song*) into more literal expressions (e.g. *final performance*) to aid both humans and machines to understand their meanings. Importantly, we perform this in a *fully unsupervised* way: our method uses only monolingual corpus data and off-the-shelf pre-trained masked language models (MLMs), and does not make use of any labelled data or lexical resources such as WordNet (Fellbaum, 1998), in contrast with previous work (Liu and Hwa, 2016; Zhou et al., 2021, 2022). We base our experiments on SemEval 2022 Task 2 (Tayyar Madabushi et al., 2022), a task designed to evaluate how well models understand the semantics of MWEs in two high-resource languages (English and Portuguese) and one low-resource language (Galician). We show that our model generates high-quality paraphrases of MWEs and aids pre-trained models to produce better representations for sentences that contain idiomatic expressions. Compared to system submissions to the shared task, our method performs better than all unsupervised systems and comparably with supervised systems.

2 Methodology

Given a target sentence t that contains an MWE x, our goal is to paraphrase x with more literal expressions in the context of t. To this end, we propose a fully unsupervised method which employs monolingual corpus data and an off-the-shelf masked language model as is, i.e. without any fine-tuning. Our method consists of the following five steps: (1) collect sentences containing x from a monolingual corpus; (2) cluster the sentences; (3) generate paraphrases of x for each cluster; (4) rerank the paraphrase candidates; and (5) select the most relevant cluster to the sense of x in the target sentence t, and paraphrase x. Figure 1 illustrates the overview of our model, and we describe the details of each step below.

2.1 Sentence Retrieval and Clustering

We first collect sentences that contain the target MWE x from a monolingual corpus.[3] Next, we sparsify sentences that have very similar local con-

*This work was partially done when the first author was at Riken.

[1]Code is available at: https://github.com/twadada/mwe-paraphrase.

[2]The MWE is said to originate from an ancient legend that a swan sings beautifully before it dies.

[3]Here, we do not perform lemmatisation of x and regard, say, *ghost town* and *ghost towns* as different instances because we aim to generate paraphrases that fit well both syntactically and semantically in the target sentence t.

Findings of the Association for Computational Linguistics: ACL 2023, pages 4732–4746
July 9-14, 2023

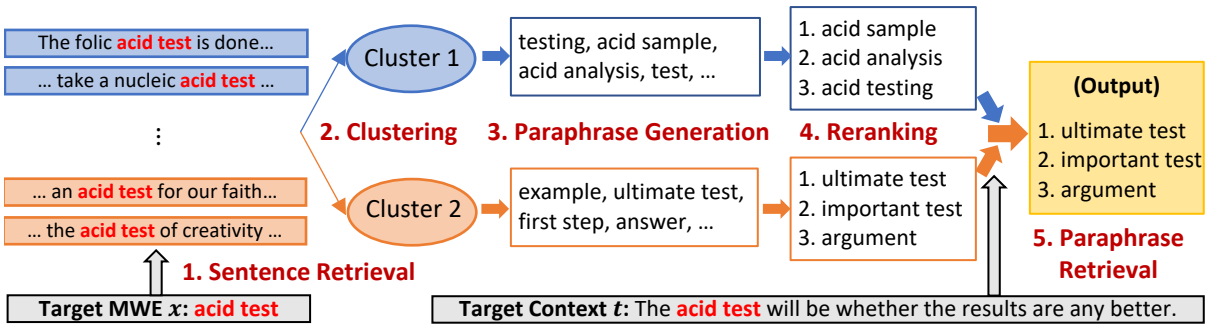

Figure 1: Overview of our proposed method.

texts around x to ensure diversity in the data,[4] and keep up to 300 sentences $\{s_i\}$ for each MWE type. Then, we generate contextualised embeddings of x for each sentence and cluster them to try to separate out different senses of x. For instance, for $x =$ *closed book*, there should be at least two clusters to represent its literal sense ("unopened book") and idiomatic sense ("mystery"; e.g. *This subject is a closed book to me*). To generate the embeddings, we replace x with a single [MASK] token and obtain the representation of [MASK] just before the linear layer prior to the word prediction output.

As our clustering algorithm, we choose DBSCAN (Ester et al., 1996) since it adaptively determines the number of clusters (expected to roughly correspond to the number of senses of x), and dynamically removes outliers during clustering. We measure the distance using cosine similarity, and tune a couple of hyperparameters of DBSCAN on the dev set of the SemEval STS task.[5] In Section 4.1, we show that DBSCAN is more effective than two other popular clustering methods: K-means (Lloyd, 1982; Arthur and Vassilvitskii, 2007) and X-means (Pelleg and Moore, 2000). As monolingual corpora, we use OSCAR (Ortiz Suárez et al., 2020) for English and Portuguese, and CC-100 (Wenzek et al., 2020; Conneau et al., 2020) for Galician.

2.2 Paraphrase Generation

Given the N_j sentences in the j-th cluster S_j: $s_{1,j}, s_{2,j}.., s_{N_j,j}$, we generate paraphrase candidates of x. The key idea is to find words or phrases that are a good fit for the contexts of x in all sentences in S_j. To this end, we use BERT to generate 1- and 2-token paraphrases, independently. To predict 1-token candidates y, we simply replace x in $s_{i,j}$ with a single [MASK] token (denoted as M), and obtain the probability as:

$$\mathrm{P}_1(y|S_j) = g(\frac{1}{N_j}\sum_{i=1}^{N_j} f(s_{i,j}^{x\to \mathrm{M}})_{\mathrm{M}}), \quad (1)$$

where $s_{i,j}^{x\to \mathrm{M}}$ denotes the sentence $s_{i,j}$ with x replaced by [MASK]; $f(s_{i,j}^{x\to \mathrm{M}})_{\mathrm{M}} \in \mathbb{R}^d$ denotes the [MASK] representation before the last linear layer; and g denotes the last linear layer followed by the softmax function. We calculate the probability distribution after averaging the [MASK] embeddings over the N_j sentences, and retrieve the top-10 words as the 1-token paraphrase candidates.

Similarly, when we generate 2-token paraphrases y: y_1y_2, we replace x with two [MASK] tokens (denoted as $\mathrm{M_1M_2}$) and obtain the probability distribution for y_k as:

$$\tilde{\mathrm{P}}(y_k|S_j) = g(\frac{1}{N_j}\sum_{i=1}^{N_j} f(s_{i,j}^{x\to \mathrm{M_1M_2}})_{\mathrm{M_k}}). \quad (2)$$

Since extracting the most probable words independently from $\tilde{\mathrm{P}}(y_1|S_j)$ and $\tilde{\mathrm{P}}(y_2|S_j)$ does not always result in a valid phrase, we first extract the top-5 words in $\tilde{\mathrm{P}}(y_1|S_j)$ and use each of them as the basis of $\mathrm{M_1}$, and then compute the probability of filling $\mathrm{M_2}$ given y_1, as:

$$\tilde{\mathrm{P}}(y_2|y_1, S_j) = g(\frac{1}{N_j}\sum_{i=1}^{N_j} f(s_{i,j}^{x\to \mathrm{y1M_2}})_{\mathrm{M_2}}). \quad (3)$$

We extract the top-5 words for each of the top-5 y_1 candidates, resulting in 25 unique phrases, some of which are single words consisting of two subword tokens. The (normalised) joint probability of y: y_1y_2 is estimated as:

$$\mathrm{P}_2(y|S_j) = \sqrt{\tilde{\mathrm{P}}(y_2|y_1, S_j)\tilde{\mathrm{P}}(y_1|S_j)}. \quad (4)$$

[4] We look at 3 words surrounding x for each sentence, and discard sentences that share many words with other sentences.

[5] The tuned values are shown in Appendix A.

Lastly, we swap the mask-filling order of y_1 and y_2 and generate another 25 phrases, some of which overlap with the previous 25 phrases. We retain the top-10 paraphrases with the largest joint probabilities and combine them with the top-10 single-token paraphrases.[6] While this algorithm could be extended to generating longer phrases, we focus on 1- and 2-token paraphrases based on the observation that many MWEs can be regarded as single semantic units (Chafe, 1968) and are thus paraphrasable with single words (e.g. *kick the bucket* with *die*) (Baldwin and Kim, 2010). In Section 3.2, we also experiment with using T5 (Raffel et al., 2020) instead of BERT to generate paraphrases without the token-length constraint.

2.3 Outer Probability Reranking

After generating paraphrase candidates with differing token lengths, the question is how to jointly rank them. One simple solution is to compare the mask-filling probabilities $\mathrm{P}_1(y|S_j)$ and $\mathrm{P}_2(y|S_j)$ directly. However, these values are not directly comparable because the former tends to get higher values due to the narrower search space. Also, these generation probabilities are affected by the word frequency of y, with rare words receiving smaller values.[7] As such, we propose a new reranking method based on the "outer probability": the probability of generating the *context* of x given the candidate. More specifically, for each sentence $s_{i,j}$ in the cluster j, we first replace some context words in $s_{i,j}$ with [MASK] tokens to create the masked sentence $\hat{s}_{i,j}$. Then, we replace x with one of the paraphrase candidates y and predict the masked tokens using an MLM. Our hypothesis is that if y represents the semantics of x very well, the model will predict the surrounding words with higher probabilities.[8] Lastly, we calculate the reranking score $\mathrm{S}(y|S_j)$ by taking the average of the log probabilities over all the masked tokens in all the sentences $s_{i,j}$ in cluster j:

$$\mathrm{S}(y|S_j) = \sum_{i=1}^{N_j} \log \mathrm{P}(s_{i,j}^{x \to \mathrm{y}} | \hat{s}_{i,j}^{x \to \mathrm{y}}),$$

where $\mathrm{P}(s_{i,j}^{x \to \mathrm{y}} | \hat{s}_{i,j}^{x \to \mathrm{y}})$ is the product of the probabilities of reconstructing the masked tokens. One advantage of this method is that the tokens we predict are always the same regardless of y, reducing the influence of the number of tokens in y.

Regarding which tokens to mask in $s_{i,j}$, one naive approach is to randomly mask words as performed during the MLM pre-training. Ideally, however, we want to select words that are semantically relevant to x; for instance, given the context: *This show will serve as his swan song, as he plans to retire*, the words *show* and *retire* are arguably more relevant to the meaning of *swan song* than other words such as *will* and *plans*. With this in mind, we mask words based on the self-attention weights: first, we replace x with two [MASK] tokens and calculate their self-attention weights from the other words in the last layer.[9] We then mask the top-5 words with the highest weights, excluding punctuation and subword tokens.

2.4 Paraphrase Retrieval

Given the ranked paraphrase candidates for each cluster, we finally retrieve the paraphrases of x by retrieving the cluster that best represents x in the target sentence t. To this end, we first replace x in t with one [MASK] token and retrieve the closest cluster based on the cosine similarity between the [MASK] embedding and the centroids of the clusters. Note that all the previous steps can be done without t, meaning if we have a list of potential MWE types and pre-compute their paraphrase candidates, we can paraphrase them given an arbitrary context in an online manner.

3 Experiments

3.1 Idiomatic Semantic Text Similarity Task

3.1.1 Data

We first evaluate our model on SemEval 2022 Task 2 Subtask B (Tayyar Madabushi et al., 2022). This is a variant of the semantic textual similarity (STS) task, where given two input sentences, a model produces a score between 0.0 and 1.0 based on

[6] We discard candidates that are highly similar to the target MWE (e.g. *swan songs* for *swan song*) based on a threshold on the normalised character-level edit distance between the candidate and x of 0.2 or less.

[7] For instance, Lau et al. (2020) measure the acceptability of a sentence based on the LM perplexity and show that it is crucial to normalise the perplexity by unigram probability of the words in the sentence because rare words tend to receive low probabilities even if they are used in a natural context.

[8] This resembles the training objective of skip-gram (Mikolov et al., 2013).

[9] We average the weights across all attention heads and [MASK] tokens. We also tried using only one [MASK] token instead of two and obtained similar results.

the similarity of the sentences, and the evalation is based on Spearman's rank correlation to the human-annotated scores. One of the input sentences E contains an MWE (e.g. E: *Witten's swan song was far from a hit*) and the other sentence is a replica of E except that the MWE is replaced with either a correct paraphrase ($E_{\rightarrow c}$, e.g. *Witten's final performance was far from a hit*) or an incorrect one ($E_{\rightarrow i}$, e.g. *Witten's bird song was far from a hit*). The target MWEs are two-word nominal compounds that can contain adjectives (e.g. *old flame*). For E–$E_{\rightarrow c}$ pairs, the STS score is 1.0 since they have almost identical meaning, while for E–$E_{\rightarrow i}$ pairs, sentences are scored in the $[0, 1)$ range. Data is provided in English, Portuguese, and Galician, and for English and Portuguese, general STS benchmark data sets — STS Benchmark (Cer et al., 2017) for English and ASSIN2 STS (Real et al., 2020) for Portuguese — are also included in the evaluation set to assess the model's generalisability on both MWE and non-MWE data.

3.1.2 Models

There are two settings for the shared task: Fine-Tune and Pre-Train. In the Fine-Tune setting, systems can be supervised on the train split of the MWE STS data,[10] but in the Pre-Train setting, systems are not allowed to use this data but can be pre-trained on other resources. Since our model works without any labelled data (or even fine-tuning), we compare our model against Pre-Train systems.

In the Pre-Train setting, the best-performing system (Phelps, 2022) expands the BERT vocabulary and obtains additional embeddings for each MWE type using monolingual data containing the MWEs. As such, it is built on similar data to our model. To obtain the MWE embeddings, they employ BERTRAM (Schick and Schütze, 2020), which was originally proposed for learning additional BERT input embeddings of rare words. The model with the MWE embeddings is then fine-tuned on the train split of *general* STS data. The major limitation of this approach is that the MWE embeddings need to be pre-trained for each model (e.g. BERT-base, BERT-large, T5) before they are fine-tuned. In contrast, our model *directly paraphrases the input text* E containing an MWE, and we feed it to an arbitrary pre-trained STS model to generate the sentence embedding of E, which is then used to measure the similarity of the sentence pairs.[11]

[10]For Galician, there is no train or development data.

[11]The other input text ($E_{\rightarrow c}$ or $E_{\rightarrow i}$) is fed into the STS model without any paraphrasing.

Model	All	MWE	General
Unsupervised			
Sem-Base	48.10	22.63	83.11
Phelps (2022)	64.02	40.30	86.41
OURS	65.31	42.65	**86.91**
OURS-ave3	**66.13**	**42.68**	**86.91**
Supervised			
Sem-Base	59.51	39.90	59.61
Phelps (2022)	65.04	41.24	81.88
Liu et al. (2022)	66.48	42.77	66.37

Table 1: Results (Spearman's rank correlation $\times 100$) on the SemEval STS task. The best scores among the unsupervised models are boldfaced.

That is, our paraphrasing model is completely separated from the task-specific models, providing more flexibility in terms of model selection and training. In the STS experiments, however, we use the same BERT models used in Phelps (2022) for both paraphrasing and STS models (and also for clustering) for fair comparison. Specifically, we use BERT-base-uncased (Devlin et al., 2019) for English, BERTimbau-Base (Souza et al., 2020) for Portuguese, and Bertinho-Base (Vilares et al., 2021) for Galician. To obtain STS models, we fine-tune the respective BERT models on the train split of STS Benchmark for English and ASSIN2 STS for Portuguese and Galician, following Phelps (2022).[12]

3.1.3 Results

Table 1 shows the results on the STS task, where the columns "MWE" and "General" denote the Spearman's rank correlation on the MWE and general STS data, and "All" denotes the overall performance. **Sem-Base** denotes the mBERT-based baseline provided by the shared task organisers. **OURS** and **OURS-ave3** denote our models, where the latter indicates the performance when we produce the sentence embedding of E by replacing the target MWE with the top-3 paraphrases and averaging the embedddings of the three sentences. In the shared task, the systems are ranked based on the "All" performance, and the table shows that our model outperforms Phelps (2022) and achieves the best score in the Pre-Train setting. It also shows

[12]The Portuguese data is used for Galician since there is no STS Galician data set and these languages are very similar.

	Model	All	MWE	General
EN	Phelps (2022)	74.45	44.22	87.09
	+ Fine-Tune	**76.43**	48.61	83.44
	OURS-ave3	76.31	**50.43**	**88.74**
PT	Phelps (2022)	70.87	**48.06**	80.10
	+ Fine-Tune	73.07	46.43	79.08
	OURS-ave3	**73.97**	45.30	**80.54**
GL	Phelps (2022)	—	29.24	—
	+ Fine-Tune	—	28.59	—
	OURS-ave3	—	**34.74**	—

Table 2: Results on the SemEval STS task for each language. The best scores for each language are boldfaced.

that using the top-3 paraphrases further improves performance.

The "Supervised" sub-table shows the performance of the submitted systems in the Fine-Tune setting, where the models are trained on both MWE and general STS data. Without any labelled data, our model outperforms the supervised task baseline (Sem-Base) and also the supervised model of Phelps (2022), which fine-tunes their unsupervised model on the MWE STS data and ranks 2nd in the Fine-Tune setting. OURS-ave3 performs slightly worse than the best supervised system (Liu et al., 2022), which however performs very poorly on the general STS data, suggesting it is overfitting the MWE data. Table 2 compares the performance of Phelps (2022) (w/ or w/o fine-tuning) and our model for each language. Overall, our model achieves better performance than the unsupervised baseline in all languages, and the supervised model in Portuguese and Galician. We expect further improvements by fine-tuning BERT on labelled or unlabelled data, which we leave for future work.

3.2 MWE Paraphrase

Next, we evaluate our model based on the matching accuracy of the MWE paraphrases. To this end, we first extract the pairs of MWEs and their correct paraphrases in context from the English "train" split of the SemEval STS data set provided for the Fine-Tune setting. Since we do not use any portion of this data to train or tune our model, we can regard it as a pseudo test set for this task;[13] the size of each data split is shown in Table 10 in Appendix. In this task, in addition to BERT-base, we also combine our method with BERT-large (whole word masking), SpanBERT-large (Joshi et al., 2020), ALBERT-large (Lan et al., 2020) and T5 (Raffel et al., 2020).[14] Unlike BERT and its variants, T5 generates an arbitrary number of tokens conditioned on the encoder hidden states of all input tokens, making it difficult to aggregate the hidden states or probability distributions across multiple sentences as done in Eqn. (1). Therefore, we propose another simple method tailored for T5 (which is somewhat different from the method described in Section 2.2): we use it to generate 20 paraphrase candidates for each sentence $s_{i,j}$ in the cluster j[15] independently using beam search, which produces $20N_j$ paraphrases in total.[16] Then, we retain the paraphrases that contain 1 or 2 *words* (which can consist of more subword tokens)[17] and rank them based on their generated counts; for the 2-word candidates, we double the count because one-word candidates often get higher values, leading to worse results. We also try reranking the candidates using the outer probability (Section 2.3), but randomly masking 5 consecutive words for T5 (and SpanBERT) rather than the high-attention words because these models are trained to fill random *spans* of text rather than separate tokens. As a strong baseline for this task, we use GPT-3 (*davinci*) (Brown et al., 2020). As a prompt, we feed several triples randomly retrieved from the dev set, which consist of a sentence that contains an MWE, a question asking what is the most appropriate substitute for it, and the correct paraphrase. We feed as many examples as possible until they reach the max token limit (2048), which correspond to about 35 triplets. Lee et al. (2021) show that this approach outperforms BERT in lexical substitution, and we use their prompt with minor modification.[18]

Table 3 (under "Matching Accuracy") shows the results based on P@k: the proportion of instances where the gold paraphrase is included in the top-k predictions. For GPT-3, we report only P@1 because it is prompted to produce the single best paraphrase only, as we have only one gold para-

[13] We could not use the test set because the gold paraphrases are not publicly available.

[14] See the Appendix for model details (Table 9).

[15] We use the T5 encoder to create the clusters.

[16] To avoid generating lengthy phrases, we set the maximum number of tokens to generate to 10.

[17] We also tried including 3-word expressions but got worse results as most gold paraphrases are 1 or 2 words.

[18] See Appendix B for an example of the prompt.

Model	# param	Matching Accuracy			STS
		P@1	P@5	P@10	ρ
GPT-3	175B	**13.2**	—	—	74.2
BERT-base	110M	8.2	18.4	24.3	76.3
BERT-large	340M	8.1	19.4	27.6	76.1
SpanBERT	340M	8.5	22.0	28.9	76.2
ALBERT	17M	6.6	13.5	21.3	75.4
T5-base	220M	**10.8**	22.0	25.8	73.6
+ rerank		10.1	22.6	27.1	**76.9**
T5-large	770M	8.0	22.5	28.0	74.4
+ rerank		7.2	**25.6**	**33.3**	76.0

Table 3: The performance (P@k) of GPT-3 (baseline) and our models on English MWE paraphrasing and STS tasks ("# param" = model parameter size). The best scores are boldfaced.

phrase per input sentence. We can see that GPT-3 performs the best in P@1, followed by our models using T5 and BERT.[19] However, note that they are not strictly comparable in terms of the number of the model parameters (shown as "# param") as well as the amount of training data (with BERT trained on the least data). Also, GPT-3 is supervised on labelled data while our models are fully unsupervised.[20] In P@$5/10$, T5-large performs the best, but using much more parameters than the other models. Our reranking method improves the performance of T5 overall, demonstrating its effectiveness. BERT performs reasonably well at P@10 with fewer parameters, possibly because BERT is conditioned on all sentences $s_{i,j}$ in the cluster j simultaneously by averaging the mask embeddings.

We also evaluate the models on the STS task (using the English test split), and the result is shown under "STS" in Table 3. Notably, GPT-3 performs the worst in this metric, and this is likely because it sometimes copies the target MWEs or paraphrases them with another MWE (e.g. *dead end* with *blind alley*), which does not simplify the text very much. The comparison of the other models also reveals that larger models are not necessarily better at text simplification.

[19]One reason why P@1 is very low is that the SemEval data contains only one gold paraphrase for each MWE (e.g. *final performance* for *swan song*), missing many other possible candidates (e.g. *last performance*, *final appearance*, and *farewell appearance*).

[20]Also, GPT-3 implicitly memorises the definitions of some MWEs during pre-training, e.g. given a simple prompt like *What is the meaning of "swan song"?*, it returns its origin as well as its meaning.

Method	P@k (EN)			P@k (PT)		
	1	5	10	1	5	10
None	6.5	15.8	**24.4**	9.7	15.9	**21.1**
K-means (2)	7.7	16.5	23.2	9.8	16.4	18.5
K-means (3)	7.2	16.0	22.5	9.9	15.9	18.4
K-means (4)	6.2	13.8	21.0	10.1	15.2	16.5
X-means	7.8	16.4	24.3	9.5	16.9	19.0
DBSCAN	**8.2**	**18.4**	24.3	**10.6**	**18.3**	20.9

Table 4: The performance of our approach with different clustering methods on the MWE paraphrasing task.

4 Analysis

4.1 Effects of Clustering

We compare the performance of our model (BERT-base) using different clustering methods: K-means, X-means and DBSCAN. Table 4 shows the results on the MWE paraphrasing task. The first row "None" denotes the performance when we treat all sentences as a single cluster. Amongst all methods, DBSCAN performs the best at P@$1/5$, outperforming K-means and X-means. We conjecture that DBSCAN benefits from not only setting the cluster size per MWE type dynamically, but also from creating one outlier cluster and discarding sentences that do not provide sufficient context to infer the MWE semantics (which is not achieved by K/X-means). Without clustering, our method performs poorly at P@$1/5$, and yet is equivalent to DBSCAN at P@10. This is because our model with clustering completely fails when it retrieves a wrong cluster,[21] whereas our model without clustering always generates a mixed bag of paraphrases with varying meanings, some of which are often relevant to the (common) MWE senses.

Table 5 shows some examples of sentences[22] sampled from each DBSCAN cluster, as well as the best paraphrase generated for each cluster. The first sentences for each MWE are "outlier sentences" sampled from the outlier cluster, and are thus not used for paraphrase generation. We can see that the outlier sentences of *inner circle* and *small fry* have vague contexts that do not represent the MWE semantics very well. In contrast, the one of *silver bullet* is regarded as an outlier despite its specific context because the target MWE is conjoined with another MWE (*magic pill*) and the model wrongly

[21]This is particularly pronounced in the low performance of K-means ($K = 4$) in P@10.

[22]The input text fed to the clustering model usually contains the previous and/or following sentences as well.

Top-1 paraphrase	Sampled sentences from each cluster
–	There is no magic pill or **silver bullet** because each of us is different.
perfect solution	But it isn't a **silver bullet** and drawbacks exist.
single practical	There is no **silver bullet** machine learning algorithm that works well across all problem spaces.
–	You don't even need to be **closed book** as you do the practice exam questions, if you don't feel ready to.
book	Following my prayer, I held the **closed book** in my hands and turned to today's passage in the day book.
complete mystery	The answer to this question is still a **closed book** for some modern historians.
fully written	Complete the **closed book** exam without removing questions or answers.
–	Does anyone in the **inner circle** know if he's ok?
personal staff	What good leaders do is recognize talent and choose their **inner circle** and cabinet based on that knowledge.
small group	I am very picky about who I let into my **inner circle** of friends for this very reason.
array	Developing an **inner circle** of great thinkers within the organization you consider your partners.
–	This may seem like a **small fry** but it isn't.
nothing special	It just seems like **small fry** compared to the struggles of last century.
young person	and i'm just a **small fry** in society.
small metal	Add it to a **small fry** pan with some cooking spray to heat and brown it a bit.

Table 5: The top-1 predicted paraphrases and example sentences (with MWEs shown in bold font) for each cluster generated by our model using DBSCAN. The first sentence for each MWE is sampled from the outlier cluster, which is discarded and not used for paraphrase generation (and thus no paraphrase is given).

predicts *pill* and *drag* as substitutes for *silver bullet*.

Comparing the three clusters of *closed book* and their paraphrases, we can see that the paraphrases generated by the first and second clusters successfully distinguish its literal and idiomatic senses ("book" and "complete mystery"), but the paraphrase of the last cluster (*fully written*) does not retain the original meaning; in fact, it is very challenging to find a good (short) substitute for *closed book* in this context, as an inherent limitation of a paraphrasing approach. The table also shows that some clusters are formed based on *syntactic* constraints rather than the semantics of the MWEs; e.g. the clusters of *silver bullet* are formed based on whether the MWE is used as a noun or adjective. The last cluster of *inner circle*, which has similar semantics to the second cluster, is also created as a result of the local morphophonetic effect, where most of the sentences in this cluster include the MWE as ***an*** *inner circle of* and *all* the paraphrase candidates start with a vowel sound (e.g. *entire army*, *elite group*, *alliance*) due to the article–noun agreement effect. A similar result is observed by Wada et al. (2022) in lexical substitution, where they find that MLM predictions and representations are highly affected by such morphophonetic or morphosyntactic biases. Interestingly, the last cluster of *small fry* is composed of the sentences where *fry* is used as a component word of the other MWE *fry pan*, not of *small fry*. Other similar cases include *high life* ("expensive lifestyle") used as *high life expectancy* and *bad hat* ("troublemaker") used as *bad hat hair*. We further discuss this problem in Section 7.

Model	Method	P@k (EN)			P@k (PT)		
		1	5	10	1	5	10
BERT-b	None	**9.6**	16.8	21.2	7.6	13.5	19.2
	Rand	6.9	17.1	**25.2**	**10.8**	17.6	20.0
	Attn	8.2	**18.4**	24.3	10.6	**18.3**	**20.9**
BERT-l	None	**11.6**	21.3	26.3	10.1	17.7	**25.9**
	Rand	6.9	**21.8**	**29.3**	11.0	21.1	23.6
	Attn	8.1	19.4	27.6	**11.3**	**21.6**	23.8

Table 6: The MWE paraphrasing performance of our approach using BERT-base/large and different reranking methods. "Rand" is the average score over 3 runs.

4.2 Effects of Reranking

In Table 3, we showed that both T5-base and T5-large benefit from reranking. To further verify its effectiveness, we compare the performance of BERT-base/large w/ or w/o reranking in Table 6. The row "None" denotes the performance when we rank the candidates based on the mask-filling probabilities $\mathrm{P}_1(y|S_j)$ and $\mathrm{P}_2(y|S_j)$ in Eqn. (1) and Eqn. (4), and "Rand" and "Attn" show the performance with our reranking method, wherein we mask five words randomly or based on self-attention weights; for Rand, we average the performance over three runs. Our reranking method improves the P@$5/10$ performance of BERT-base in both English and Portuguese, but for BERT-large, there is no clear winner. When we compare Rand and Attn, there is no

Method	ALL		EN		PT		GL
	All	MWE	All	MWE	All	MWE	MWE
dev							
None	80.61	39.41	86.26	49.99	78.07	37.83	—
Rand	80.14	37.65	86.08	48.97	77.18	34.80	—
Attn	**81.24**	**41.75**	**86.70**	**51.86**	**78.31**	**38.28**	—
test							
None	65.39	41.48	75.61	49.21	73.76	44.30	33.98
Rand	65.41	41.42	75.93	49.49	73.93	**45.44**	32.19
Attn	**66.13**	**42.68**	**76.31**	**50.43**	**73.97**	45.30	**34.74**

Table 7: The STS performance (Spearman's rank correlation$\times$100) of our models (OURS-ave3) using different reranking methods. The scores of Rand are averaged over 3 runs. The best scores in each data split are shown in bold. The general STS scores are omitted from the table as they stay the same across all models (86.93/92.79 on the dev set and 88.74/80.54 on the test set for EN/PT).

noticeable difference in matching accuracy; therefore, we also examine the impact of our reranking methods on the STS task in Table 7 (which can measure the semantic fit of the paraphrases in a continuous manner). It shows that Attn performs the best overall on the dev and test sets. We also observe that the performance of Rand varies greatly across three runs (65.34, 65.91, and 64.96 in ALL on the test set), suggesting that the selection of the context words to reconstruct in our reranking method has a non-trivial impact on performance.

Table 8 shows examples of MWEs and their top-3 paraphrases generated by BERT-base w/ or w/o reranking (Attn). It shows that reranking produces semantically more relevant paraphrases. Furthermore, these results reveal a few plausible limitations of the scoring method based on the mask-filling probabilities; the limitations that are also relevant to T5 to some degree. Firstly, it tends to favour single-token paraphrases (e.g. *bridge* and *road* for *zebra crossing*) due to the narrower search space. Secondly, it often assigns high scores to phrases that have strong collocation (e.g. *new world*, *regular customer*) regardless of their semantic fit because of the higher probability $\tilde{\mathrm{P}}(y_2|y_1, S_j)$ in Eqn. (3) (or $\tilde{\mathrm{P}}(y_1|y_2, S_j)$). Thirdly, it is inevitably influenced by word frequency, as evidenced by the fact that *research organization* gets much higher values than its British spelling *research organisation* (0.33 vs. 0.09) as the paraphrase of *think tank*; on the other hand, our reranking method assigns them very similar scores. On the other hand, one limitation of our reranking method is that it occasionally gives high scores to syntactically ill-formed candidates, especially those that contain duplicated tokens (e.g. *clock clock* as the paraphrase for *grandfather clock*), and we alleviate this by removing one of the duplicated tokens.

MWE	w.o. Rerank	w. Rerank
zebra crossing	bridge pedestrian bridge road	**pedestrian crossing** pedestrian bridge road crossing
melting pot	new world unique mix diverse mix	**mixture** unique mix collection
think tank	research organization organization **research group**	research organization research organisation **research group**
busy bee	**busy person** good person regular customer	**busy person** busy woman busy man

Table 8: Examples of the top-3 paraphrases predicted by our model w/ or w/o reranking (Attn). The gold paraphrases are boldfaced.

5 Related Work

There is a line of work on MWE paraphrasing (or substitution), but unlike our method, previous methods resort to either human-annotated corpora or high-coverage dictionaries to generate paraphrases; consequently, they are only evaluated in English or other high-resource languages (e.g. Chinese). For instance, Liu and Hwa (2016) extract paraphrases of MWEs from their dictionary definitions,

which usually contain supplementary information as well as their core meanings and paraphrases. Similarly, Zhou et al. (2022) encode definitions of MWEs using a sentence embedding model and employ the embeddings to generate their paraphrases. They also propose another method that fine-tunes BART (Lewis et al., 2020) on a parallel corpus built by Zhou et al. (2021), in which the source sentences contain MWEs and the target sentences paraphrase them while leaving the other words unchanged. Similarly, Qiang et al. (2022) create analogous data in Chinese and fine-tune mT5 (Xue et al., 2021) on it to paraphrase Chinese MWEs. In contrast, Ponkiya et al. (2020) propose an unsupervised method using BERT/T5, but they focus on paraphrasing noun compounds (e.g. *club house*) using the same MWE tokens (e.g. *house owned by a club*) without context.

A more common and resource-efficient approach to handling MWEs is to regard them as individual lexical units (e.g. regard "kick_the_bucket" as one token) and train their embeddings using monolingual data (Salehi et al., 2015; Cordeiro et al., 2019; Phelps, 2022). Those embeddings can be used to retrieve similar words in the vocabulary based on embedding similarity (Otani et al., 2020). However, one drawback is that it increases the size of vocabulary (and parameters) significantly, due to the sheer volume of MWE instances (e.g. around 41% of entries in WordNet 1.7 (Fellbaum, 1998) are MWEs (Sag et al., 2002)). Recently, Zeng and Bhat (2022) addressed this limitation by training an additional adapter network (Pfeiffer et al., 2020) on top of an MLM to produce better embeddings for various MWEs, but they rely on dictionary definitions to train the network, arguing that such external knowledge is fundamental for learning MWE representations.

6 Conclusion

We propose a fully unsupervised method to paraphrase multiword expressions (MWEs) in context. Our method employs only a monolingual corpus and pre-trained language model, and does not rely on any labelled data. In our experiments, we show that our model generates good MWE paraphrases and aids pre-trained sentence embedding models to represent sentences containing MWEs.

7 Limitations

One limitation of our proposed method is that it requires the pre-identification of the target MWE in a sentence before paraphrasing it, a task that is not a walk in the park. In particular, it is very challenging to identify what is the "correct" span of a given MWE, which our model critically relies on. For instance, given the MWE *lip service* ("insincere agreement"), our model predicts *more attention* as the best paraphrase, likely because the MWE is usually used as *pay lip service to (something)*, and *attention* is one of the few words that fits well in this context (in terms of collocation). Therefore, the whole phrase *pay lip service to* should be identified as an MWE instance[23] when it is used in sentences like *They pay lip service to the idea*; however, *lip service* can also serve as one lexical unit in sentences like *It wasn't just lip service*. A similar problem arises when we deal with nominal MWEs that follow indefinite articles (*a* or *an*) as discussed in Section 4.1, or verbal MWEs that are often followed by specific prepositions (e.g. *turn a blind eye **to** ...* means "deliberately ignore ...") because the MLM prediction is affected by the syntactic constraint.[24] MWE span identification is also important in our sentence collection process; e.g. as discussed in Section 4.1, the phrase *small fry* can be used as *small fry pan* rather than as the MWE meaning "insignificant", and hence collecting sentences based on string match resulted in one additional cluster that is not relevant to either its literal or idiomatic senses.

Another limitation is that our model cannot handle discontinuous MWEs such as *throw someone under the bus* and *not ... in the least* because it is not clear which parts to mask and paraphrase in such cases. Similar problems arise when continuous MWEs undergo either internal modification (e.g. *go completely cold turkey*) or drastic syntactic transformation (e.g. *the beans are split*). However, note that all of these types of expressions, as well as the pre-tokenisation problem discussed above, become a pain in the neck for any approach that regards an MWE as a lexical unit and learns its holistic embedding.

Lastly, our method heavily relies on the quality

[23] In fact, it is registered as such in some English dictionaries.

[24] In languages where words have grammatical gender such as Portuguese and Italian, this problem can be more pronounced because context words including adjectives and determiners are affected by gender.

of the clusters and is thus prone to error propagation. For instance, our model using BERT always generates *large fish* as the best paraphrase for the MWE *big fish* and fails to capture its idiomatic sense ("an important person"), likely due to its rare occurrence in monolingual corpora (compared to its literal sense). One possible solution to this problem is to derive more senses by allowing the clustering method to create more clusters with fewer instances, but that institutes a trade-off between accommodating rare senses and creating too many clusters for common senses; hence, there is no silver bullet. In fact, this problem pertains to the longstanding question (with no single correct answer) among lexicographers: how to "split" and "lump" senses of words, and how fine-grained the sense distinctions should be (Hanks, 2000, 2012).

References

Pierre J. L. Arnaud and Sandra J. Savignon. 1996. Rare words, complex lexical units and the advanced learner. In James Coady and Thomas Huckin, editors, *Second Language Vocabulary Acquisition: A Rationale for Pedagogy*, pages 157–173. Cambridge University Press.

David Arthur and Sergei Vassilvitskii. 2007. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA. Society for Industrial and Applied Mathematics.

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. In *Handbook of Natural Language Processing*, pages 267–292, Boca Raton, USA. CRC Press.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Wallace L. Chafe. 1968. Idiomaticity as an anomaly in the chomskyan paradigm. *Foundations of Language*, 4(2):109–127.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Silvio Cordeiro, Aline Villavicencio, Marco Idiart, and Carlos Ramisch. 2019. Unsupervised Compositionality Prediction of Nominal Compounds. *Computational Linguistics*, 45(1):1–57.

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.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Patrick Hanks. 2000. Do word meanings exist? *Computers and the Humanities*, 34(1/2):205–215.

Patrick Hanks. 2012. The Corpus Revolution in Lexicography. *International Journal of Lexicography*, 25(4):398–436.

Suzanne Irujo. 1986. Don't put your leg in your mouth: Transfer in the acquisition of idioms in a second language. *TESOL Quarterly*, 20(2):287–304.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. 2020. How furiously can colorless green ideas sleep? sentence acceptability in context. *Transactions of the Association for Computational Linguistics*, 8:296–310.

Mina Lee, Chris Donahue, Robin Jia, Alexander Iyabor, and Percy Liang. 2021. Swords: A benchmark for lexical substitution with improved data coverage and quality. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4362–4379, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Changsheng Liu and Rebecca Hwa. 2016. Phrasal substitution of idiomatic expressions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 363–373, San Diego, California. Association for Computational Linguistics.

Kuanghong Liu, Jin Wang, and Xuejie Zhang. 2022. YNU-HPCC at SemEval-2022 task 2: Representing multilingual idiomaticity based on contrastive learning. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 211–216, Seattle, United States. Association for Computational Linguistics.

S. Lloyd. 1982. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations (Workshop)*, Scottsdale, Arizona, USA.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Naoki Otani, Satoru Ozaki, Xingyuan Zhao, Yucen Li, Micaelah St Johns, and Lori Levin. 2020. Pretokenization of multi-word expressions in cross-lingual word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4451–4464, Online. Association for Computational Linguistics.

Dan Pelleg and Andrew W. Moore. 2000. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 727–734, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Dylan Phelps. 2022. drsphelps at SemEval-2022 task 2: Learning idiom representations using BERTRAM. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 158–164, Seattle, United States. Association for Computational Linguistics.

Girishkumar Ponkiya, Rudra Murthy, Pushpak Bhattacharyya, and Girish Palshikar. 2020. Looking inside noun compounds: Unsupervised prepositional and free paraphrasing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4313–4323, Online. Association for Computational Linguistics.

Jipeng Qiang, Yang Li, Chaowei Zhang, Yun Li, Yunhao Yuan, Yi Zhu, and Xindong Wu. 2022. Chinese idiom paraphrasing.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Livy Real, Erick Fonseca, and Hugo Gonç alo Oliveira. 2020. The ASSIN 2 shared task: A quick overview. In *Computational Processing of the Portuguese Language - 14th International Conference, PROPOR 2020, Evora, Portugal, March 2-4, 2020, Proceedings*, volume 12037 of *Lecture Notes in Computer Science*, pages 406–412. Springer.

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.

Bahar Salehi, Paul Cook, and Timothy Baldwin. 2015. A word embedding approach to predicting the compositionality of multiword expressions. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 977–983, Denver, Colorado. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. BERTRAM: Improved word embeddings have big impact on contextualized model performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3996–4007, Online. Association for Computational Linguistics.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2020. BERTimbau: Pretrained BERT models for Brazilian Portuguese. In *Intelligent Systems*, pages 403–417, Cham. Springer International Publishing.

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.

Harish Tayyar Madabushi, Edward Gow-Smith, Carolina Scarton, and Aline Villavicencio. 2021. AStitchInLanguageModels: Dataset and methods for the exploration of idiomaticity in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3464–3477, Punta Cana, Dominican Republic. Association for Computational Linguistics.

David Vilares, Marcos García, and Carlos Gómez-Rodríguez. 2021. Bertinho: Galician BERT representations. *Procesamiento del lenguaje natural*, 66:13–26.

Takashi Wada, Timothy Baldwin, Yuji Matsumoto, and Jey Han Lau. 2022. Unsupervised lexical substitution with decontextualised embeddings. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4172–4185, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Ziheng Zeng and Suma Bhat. 2022. Getting BART to Ride the Idiomatic Train: Learning to Represent Idiomatic Expressions. *Transactions of the Association for Computational Linguistics*, 10:1120–1137.

Jianing Zhou, Hongyu Gong, and Suma Bhat. 2021. PIE: A parallel idiomatic expression corpus for idiomatic sentence generation and paraphrasing. In *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, pages 33–48, Online. Association for Computational Linguistics.

Jianing Zhou, Ziheng Zeng, Hongyu Gong, and Suma Bhat. 2022. Idiomatic expression paraphrasing without strong supervision. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11774–11782.

A Hyper-Parameters

Table 9 shows the hyper-parameters of DBSCAN: minPts: the minimum number of points required to form a core point; and ϵ: the maximum distance between two points to be considered as neighbours. We tune ϵ for each model, and the table shows their shortcut names in the Transformers library (Wolf et al., 2020). For SpanBERT, we used the model in the original GitHub repository (`https://github.com/facebookresearch/SpanBERT`).

B GPT-3 Prompt

Here is an example of the GPT-3 prompt used in Section 3.2:

```
Witten's **swan song** was
far from a hit.

Q: What is the most
appropriate substitute
for **swan song** in the
above text?

A: final performance
```

It is a triple of the target sentence with the target MWE marked with **, a question that asks the most approproate substitute for the MWE, and the gold paraphrase retrieved from the dev set. We borrow this template from Lee et al. (2021), but

Parameters	Model	Value
minPts	—	$\max(3, \lfloor 0.03N \rceil)$
ϵ	bert-base-uncased	0.4
	bert-large-uncased-whole-word-masking	0.5
	spanbert-large-cased	0.3
	albert-large-v2	0.3
	google/t5-v1_1-base	0.4
	google/t5-v1_1-large	0.4
	neuralmind/bert-base-portuguese-cased	0.3
	neuralmind/bert-large-portuguese-cased	0.3
	dvilares/bertinho-gl-base-cased	0.3

Table 9: The hyper-parameters of DBSCAN for each model tuned on the STS dev set. N denotes the number of the sampled sentences.

Split	# MWE types			# Sents with MWEs		
	EN	PT	GL	EN	PT	GL
train	214	108	0	4,725	1,847	0
dev	30	20	0	521	454	0
test	50	50	50	1,419	1,124	1,367

Table 10: The numbers of the MWE types and target sentences that contain them in each STS data set. The train split is used as a pseudo test set for the MWE paraphrasing task in Section 3.2.

change *What are appropriate substitutes* to *What is the most appropriate substitute* since we have only one gold paraphrase for each input text. We feed as many triples as possible until they reach the max token limit (2048), which correspond to about 35 triplets. Then, we append one "test triplet" that contains one sentence from the test set, the corresponding question, and the answer without the gold paraphrase (i.e. *A:*), and make the model predict the paraphrase. We make sure that the MWE in the test triplet is not included in the triplets retrieved from the dev set.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Left blank.

☑ B1. Did you cite the creators of artifacts you used?
3

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We use the data from the SemEval 2022 shared task, and we follow its intended use.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
3

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☑ Did you run computational experiments?

3, 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
2, 3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.