Paraphrase Identification: Lightweight Effective Methods Based Features from Pre-trained Models

Abu Bakar Siddiqur Rahman[1,3,*], Hoang Thang Ta[1,2], Lotfollah Najjar[3] and Alexander Gelbukh[1]

[1]*Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC), Mexico City, Mexico*
[2]*Dalat University, Lam Dong, Vietnam*
[3]*University of Nebraska Omaha, Omaha, Nebraska, USA*

Abstract

In this paper, we work on Paraphrase Identification in Mexican Spanish (PAR-MEX) at the sentence level. We introduced two lightweight methods, linear regression and multilayer perceptron for training data on features, extracted from pre-trained models. A rule of thumb, pair similarity is used to filter noises in the positive examples. We obtained the best F1 of 88.67%, which points out the effectiveness of traditional methods with the support of pre-trained models. In the challenge, our result ranked fourth in the organizers' result table.

Keywords

Paraphrase Identification, Text Classification, Linear Regression, MultiLayer Perceptron, PAR-MEX, IberLEF

1. Introduction

Paraphrase identification means identifying whether two sentences have the same meaning or not. According to classification problems, if a pair of texts is given, the model should determine the semantic similarity between two sentences. If the two sentences or two paragraphs are identical, then it is known as paraphrase; otherwise, non-paraphrase. A sentence can be pieces by word. There might be the possibility of the words in two sentences of an article being either lexically similar or semantically similar. Semantically similar refer to the same meaning of two sentences whereas lexically similar refer to the same character or word matching in the sentences. For example, Robert cooks vegetables, and vegetables cook Robert. These two sentences are lexically similar, however, not similar by semantics.

Paraphrase identification is an essential task for plagiarism detection, authorship authentication, text summarization, information retrieval, question answering, and text mining tasks. The academic field used plagiarism checkers for most writing purposes. About two decades ago,

IberLEF 2022, September 2022, A Coruña, Spain.

*Corresponding author.
abubakarsiddiqurra@unomaha.edu (A. B. S. Rahman); tahoangthang@gmail.com (H. T. Ta); lnajjar@unomaha.edu (L. Najjar); gelbukh@cic.ipn.mx (A. Gelbukh)
0000-0002-8581-0891 (A. B. S. Rahman); 0000-0003-0321-5106 (H. T. Ta); 0000-0003-3960-4189 (L. Najjar); 0000-0001-7845-9039 (A. Gelbukh)

400 colleges in the United States (U.S.) use a system that can detect plagiarism [1]. One of the important and questionable parts to evaluate a scholarly article or an assignment in academic work is to check whether the article or assignment do plagiarism by using a sentence or whole paragraph without citing the source article. The office of research integrity has an important role to investigate in this issue by helping the universities in the United States (U.S.) [2, 3]. The necessity of identifying paraphrases increasing day by day due to the abundance of texts are generated from social media such as Twitter, and Facebook. These texts are not clean and contain misspellings, and different styles of writing by using the short forms of words considering overall noisy texts [4]. Paraphrasing can be identified easily and accurately with the cleaned text. Microsoft paraphrase corpus [5] used to identify paraphrase as it contained only clean text. Short texts are another challenging task in the case of paraphrase identification due to measuring the sentence similarity with less common lexical features. If two sentences have the most lexical features in common, then it is possible that the remaining features are the same between the two sentences [6]. Based on the necessity and challenges of identifying paraphrases, recently machine learning (ML) and deep learning (DL) techniques are used by Natural Language Processing (NLP) researchers to build efficient methods to identify paraphrases from an article.

Bel-Enguix et. al. described the overview of paraphrase identification of Spanish corpora on PER MAX shared tasks in IberLEF 2022 where most of the participants use ML and DL techniques to detect paraphrase [7]. Support Vector Machine (SVM), K nearest neighbors, and maximum entropy algorithm were used to detect paraphrases where SVM performed the best on the task [8]. Logistic regression, SVM, and different neural network techniques were used to compare the results finding out which model provides better results for detecting paraphrase [9]. A double CNN model with multi-granular interaction features was used to detect paraphrases where the model is the combination of three parts the sentence analysis network, the sentence interaction model, and logistic regression on top of the model to classify the sentence based on the probability whether the sentence paraphrased or not [10].

In this paper, we used linear regression (LR) and multilayer perceptron (MLP) to detect paraphrasing by extracting features for making a vector from the features to feed into the models (LR and MLP) as an input. The task has been done in IberLEF 2022 paraphrase identification competition where we achieved the fourth position (https://codalab.lisn.upsaclay.fr/competitions/2345#results) based on the F1 score compared to other participants.

2. Related Works

Paraphrase detection is carried out by string-based approaches, corpus-based approaches, and knowledge-based approaches. Both corpus-based and knowledge-based works as a semantic similar where information gathered from corpora are used to detect paraphrases for corpora-based approaches and information gathered from the semantic network is used to find the degree of similarity between words for knowledge-based approaches.

One of the obstacles to analyzing paraphrase identification due to not having an available dataset. Microsoft Research Paraphrase Corpus (MRPC) contains 5801 sentence pairs with 67% annotations that were semantically equivalent based on the human judgment [11]. Another useful and available dataset is Quora Question Pairs (QQP) which is collected from the questions posted on Quora. To identify the same questions as paraphrasing is one of the most important tasks in any question-answering system, the same as Quora and stackoverflow [12, 13]. However, QQP has a less common lexical overlap between the sentence pairs. To overcome this, a dataset named PAWS (Paraphrase adversaries from word scrambling) was introduced by Zhang et.al. consisting of 108463 sentence pairs with more common lexical overlap [14]. Wahle. et. al. created a dataset that contained 1.5M paragraphs from arXiv, thesis, and Wikipedia articles with their paraphrases [15]. PAN-PC-09 is another available dataset that consists of 41223 text documents collected from project Gutenberg that consists of 22874 documents. The dataset has a mixture of small, medium, and large documents. Around half of the dataset contained small documents that length of 1 to 10 pages. 35% are medium with 10 to 100 pages and 15% are large with 100 to 1000 pages. 90% of the dataset is in English, the rest percentage is translated from German and Spanish [16]. PAN-PC-10 corpus consists of 64558 artificial and 4000 simulated plagiarism [17]. Plagiarism is automatically inserted in the dataset of PAN-PC-11 [18]. Xu et.al. collected a corpus from Twitter consisting of 18763 sentence pairs with more than 400 distinct topics [19]. Lan et. al. collected the sentence pairs from the tweets that share the same news URL [20]. Some other corpora have been created in different languages. External plagiarism detection on Arabic corpus was collected from a corpus of contemporary Arabic and Arabic wikipedia [21]. In this paper, a Spanish corpus is used from PAR-MEX 2022. From all corpora, it can be noticed that there needs to generate similar sentences for making a corpus. Paraphrases can be generated by [22, 23]. Human annotation is one of the challenging tasks to create a dataset. Word position deviation and lexical deviation metrics were used on MRPC and found out how it differs from PAWS to improve the existing dataset as a generalization case [24].

Several methods have been used for paraphrase detection. A deep neural network model by using a convolutional neural network and long short term memory network was used in [25], and different word embedding techniques such as Countvectorizer, TF-IDF Vectorizer, fastText, etc. were used in [26]. Recently, the pre-trained BERT model was used extensively in most text-based research due to its contextual representation. BERT in addition to topic models fixes the problem of representing domain-specific cases used for detecting semantic similarity [27]. Ko. et. al. used the paraphrase BERT method by fine-tuning the pre-trained BERT model and adding a whole word masking. Then multi-task learning was performed on the analysis. This results in better performance for paraphrase identification [28]. Two encoders were used with the predicate-argument structure to detect paraphrases where predicate was the verb from the sentence and argument belongs to subject, object, extension [29].

3. Task Description

The task is to detect a given pair of sentences will be a paraphrase or not, classified into paraphrase (P) and non-paraphrase (NP) correspondingly. These sentence pairs are composed manually by extracting content from more than 200 Spanish texts on 7 topics, including molecular cuisine, sushi, tequila, kebab, vegan food, food truck, and Mexican ofrenda. Organizers use 3 paraphrase methods, low paraphrase, high paraphrase, and no paraphrase.

To obtain a wide diversity, organizers modified pairs of sentences with different levels,

Table 1
The distribution of the datasets

Dataset	Total pairs	Paraphrase sentence-pairs	Non-paraphrase sentence-pairs
Training	7,382	1,282 (17.36%)	6,100 (82.63%)
Validation	97	20 (20.61%)	77 (79.38%)
Test	2,819	542 (19.22%)	2,277 (80.77%)
Total	10,298	1,844 (17.90%)	8,454 (82.09%)

from word-level by using synonyms or word orders, to sentence-level with changing sentence structure. Furthermore, a target sentence can be paraphrased by keeping only the main idea of the source sentence, without capturing all original words. Besides, a set of common words in two texts, having no relation, can be used for paraphrasing. Unfortunately, the current task is only to offer element replacement and structure modification.

As declared by organizers, this is the first edition of the task of Paraphrase Identification in Mexican Spanish (PAR-MEX). At the sentence level, the task is to identify whether a given sentence is a paraphrase of another one or not. There are 2 types of paraphrasing, high and low-level paraphrases from the only lexical overlap of the sentences. The paraphrasing process complies with the method of Torres-Moreno et al., 2014 in constructing a German corpus [30].

4. Dataset Analysis

The organizers offer us 3 sets: training set, validation set, and test set. The training set contains 7,382 sentence pairs, while 97 and 2,891 are the values of validation and test sets. The distribution of paraphrase sentence pairs (P pairs) and non-paraphrase sentence pairs (NP pairs) on all sets is approximately 8:2, as shown in Table 1. In detail, P pairs accounted for 17.35% in the training while NP pairs are 82.63%. Similarly, in the test set, the distribution of P pairs and NP pairs is 17.90% and 82.09%. In this paper, we only extract features from the dataset so we ignore the analyses on the distribution of tokens and texts. We also take the validation set out of the training due to its relatively small size.

5. Methodology

To prepare the data for LR and MLP, we extract features from sentence pairs by pre-trained models and package `es_core_news_md` of spaCy v2.3.2 (https://spacy.io/). This step is applied to both training and test sets to make sure that we have the values of test sentence pairs in the predictions.

- `text1_len`: The number of tokens of the first sentence when parsing by spaCy.
- `text2_len`: The number of tokens of the second sentence when parsing by spaCy.
- `diff_len`: The different length will be calculated between 2 sentences. We use spaCy to extract the length of 2 sentences (the number of tokens), and then get the absolute value between them by formula $|text1\_len - text2\_len|$.

- `common_lemma`: This is the common lemmas between 2 sentences. We extract lemmas of each sentence which are not stopwords, punctuations, and blanks as a set. Then, search and save the intersection of 2 sets, respectively for 2 sentences. In the training, we only use the number of common lemmas instead of their values.
- `same_aff_neg`: There is also a case where a pair of sentences has a high similarity but it is not paraphrased. This happens because both sentences are not together affirmative or negative. Unfortunately, to the best of our efforts, we did not find any available model for negation detection in Spanish in Hugging Face. We thus must apply zero-shot classification on pre-trained model `"MoritzLaurer/mDeBERTa-v3-base-mnli-xnli"` with 2 Spanish labels (`afirmación` and `negación`). Although this practice may not reflect precisely the negation in sentences, at least we have a similar feature used for the training. If both sentences are affirmative or negative, `same_aff_neg` will equal to 1. Otherwise, it will be 0. This is the same as the truth table of the XOR operator.
- `pair_sim`: This is the cosine similarity of sentence pairs, ranging from [0-1] between two sentences, calculated from theirs embeddings extracted from pre-trained model `"symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli"`, which supports the maximum length of 514 tokens (Roberta).

Finally, we store these features in independent *.json files. We realize that if a sentence pair has a paraphrase relation, its similarity is relatively high. Furthermore, its `diff_len` value is close to 0, its `same_aff_neg` value is 1 (both sentences are negative or affirmative at the same time), and it has more common lemmas than other pairs. This is a random example of a sentence pair to represent this observation.

```
{
"id": 1,
"text1": "su nombre proviene de los persas, los cuales empleaban este
 ↪  alimento para dar de comer a sus reyes y significa carne a la
 ↪  parrilla.",
"text2": "el alimento, cuyo nombre proviene de los persas, que significa
 ↪  carne a la parrilla, se empleaba para dar de comer a sus reyes.",
"label": 1,
"sim": 0.8956236908049805,
"diff_len": 1,
"text1_len": 28,
"text2_len": 29,
"same_aff_neg": 1,
"common_lemma": ["comer",  "alimentar",  "persa", "parrilla", "rey", "carne",
 ↪  "parir", "provenir", "significar", "nombrar"]
}
```

Next, we take 6 features (`text1_len`, `text2_len`, `diff_len`, `common_lemma`, `same_aff_neg`, and `pair_sim`) of each pair to build a set of vectors, $V = \{\vec{v}_1, \vec{v}_2, \vec{v}_3, ..., \vec{v}_n\}$ with $\vec{v}_i$ is a feature vector of pair i. For the visualization purpose, we then pass V as a matrix and use Principal

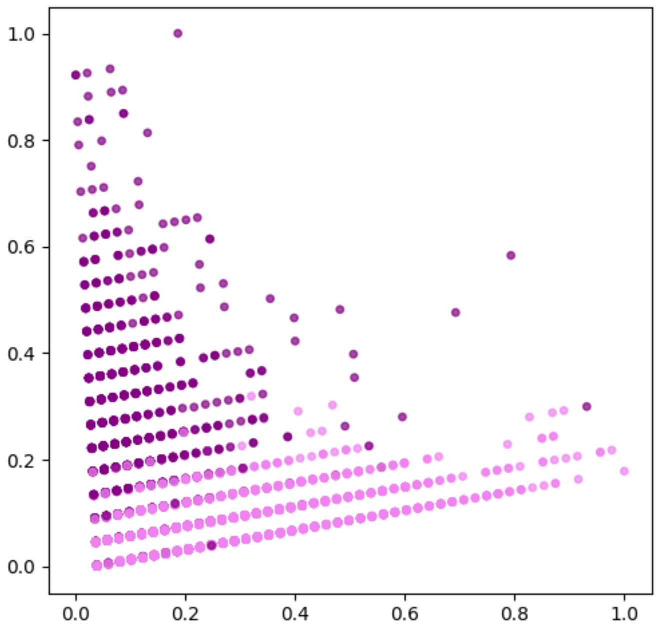

Figure 1: The distribution of sentence pairs by classes, NP and P. Dark dots are P pairs, while the light dots are NP pairs.

Component Analysis (PCA) to reduce the dimension from 6 to 2 before normalizing its values (MaxMinScale) from 0 to 1. The output matrix of the training dataset is shown in the Figure 1 with sentence pairs, colored in 2 categories. Purple (dark) dots and violet (light) dots are P pairs and NP pairs respectively. The border between 2 categories (P and NP) is intertwined and some dots of a category lie on deeply in other category and vice versa. This implies that the dataset may contain noises. To filter noises, we use a rule of thumb. A sentence pair which has `pair_sim` lower than the average value of NP category should belong to NP category. In contrast, if this pair has `pair_sim` higher than the average value of P category, it should belong to this category. For the similarity of training dataset, the average value of the whole dataset (Avg. All), the average value of NP category (Avg. NP), and the average value of P category (Avg. P) are 0.3454, 0.2504, and 0.7978 respectively. In the training, we use set V without dimensionality reduction and value normalization as the input for LR and MLP.

6. Experiment

In the experiment, there are 3 runs, shown in Table 2. MLP has 3 hidden layer sizes (128, 64, 32) with `random_rate=5`, `activation='relu'`, and `learning_rate_init=0.01`. When training with MLP, we divided the original dataset into training and validation sets with the ratio 8:2 (≈

Table 2
Our submissions' results from the organizer website.

#	Method	Noise filtering	F1*	NP Acc.	Original NP Acc.**
1	MultiLayer Perceptron	No	0.8783	0.8339	0.8077
2	Linear Regression	No	0.8837	0.8414	0.8077
3	**Linear Regression**	**Yes**	**0.8867**	**0.8329**	**0.8077**
4	*Baseline (BETO & BERT)*	-	*0.7026*	-	-

*The results taken from the organizers' website.
**According statistics of the original datasets in Table 1.

5905:1477). After 57 iterations, the best model was obtained the highest validation accuracy is 97.29%. For LR, we did not split the dataset as MLP. R squared values with and without noise filtering are 0.7011 and 0.7397. There was a minor mistake when executing noise filtering. Instead of removing only P pairs that have `pair_sim` < 0.2504 (Avg. NP), we removed all pairs, including NP pairs. Unfortunately, we ran out of time so we could not able to fix this error and tried with other runs, such as removing P pairs with `pair_sim` < 0.3454 (Avg. All). Consequently, we are not able to check whether these practices will offer better results or not.

To predict the accuracy of NP pairs, we set all labels of the test set as 0. Then, we compare our generated test set from the model with this zero test set. Due to the binary classification, there are only 2 classes, 0 for NP and 1 for P. This practices is just to hack the class distribution. Fortunately, in Table 1, organizers provided statistics about the data distribution, and the accuracy of NP pairs is 80.77%. We consider this value as the accuracy of ”original” NP pairs. This somehow helps us to measure the model performance without submitting the results to the organizers. In Table 2, all NP accuracies are higher than the original values, but not too large. The result of LR with noise filtering is the best with an F1 of 88.67%. When compared to other teams in the challenge, our best method is ranked fourth. Our methods outperformed the baseline, which was fine-tuned on BERT and BETO models.

7. Conclusion

In this paper, we apply 2 lightweight methods (LR and MLP), trained on features extracted from pre-trained models in Hugging Face for paraphrase identification at the sentence level. Firstly, spaCy is used to extract features, length of 2 texts, different lengths, and common lemmas. For pair similarity and same affirmation/negation, we use 2 pre-trained models, `"MoritzLaurer/mDeBERTa-v3-base-mnli-xnli"` and `"symanto/sn-xlm-roberta-base-snli-mnli-anli-xnl"`. Then, we used a rule of thumb to filter noises before putting the cleaned dataset into the training process. We achieved the best F1 value of 88.67% with a run of LR and noise filtering, ranked fourth compared to other teams in the challenge.

In the future, we will analyze datasets to have more right features extracted from the relations between sentence pairs for paraphrase identification. Noise filtering is also an important factor to increase the model performance that we need to take our eyes on. Furthermore, we aim to use light-weight approaches and test our methods on other datasets (MRPC) to make comparisons.

Acknowledgments

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of CONACYT, Mexico, grants 20220852 and 20220859 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico, and also show high gratitude to Holland computing center, the University of Nebraska to provide their high computing GPU resources. The authors acknowledge the support of Microsoft through the Microsoft Latin America Ph.D. Award.

References

[1] A. L. Foster, Plagiarism-detection tool creates legal quandary, Chronicle of Higher Education (2002) 37–38.

[2] C. Martyn, Fabrication, falsification and plagiarism (2003) 243–244.

[3] J. M. Hunter, Plagiarism–does the punishment fit the crime? (2006) 139–142.

[4] W. Xu, A. Ritter, C. Callison-Burch, W. B. Dolan, Y. Ji, Extracting lexically divergent paraphrases from Twitter, Transactions of the Association for Computational Linguistics (2014) 435–448.

[5] D. Das, N. A. Smith, Paraphrase identification as probabilistic quasi-synchronous recognition (2009) 468–476.

[6] T. Kajiwara, D. Bollegala, Y. Yoshida, K.-i. Kawarabayashi, An iterative approach for the global estimation of sentence similarity, PloS one 12 (2017).

[7] G. Bel-Enguix, H. Gomez-Adorno, G. Sierra, J.-M. Torres-Moreno, J.-G. Ortiz-Barajas, J. Vasquez, Overview of PAR-MEX at Iberlef 2022: Paraphrase Detection in Spanish Shared Task, Procesamiento del Lenguaje Natural 69 (2022).

[8] Z. Kozareva, A. Montoyo, Paraphrase identification on the basis of supervised machine learning techniques (2006) 524–533.

[9] J.-R. Hunt, Ethan, C. Kinares, C. Koh, A. Sanchez, F. Zhan, M. Ozdemir, Machine learning models for paraphrase identification and its applications on plagiarism detection, IEEE International Conference on Big Knowledge (ICBK) (2019) 97–104.

[10] W. Yin, H. Schütze, Convolutional neural network for paraphrase identification, Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2015) 901–911.

[11] B. Dolan, C. Brockett, Automatically constructing a corpus of sentential paraphrases, Third International Workshop on Paraphrasing (IWP2005) (2005).

[12] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. R. Bowman, GLUE: A multi-task benchmark and analysis platform for natural language understanding, arXiv preprint arXiv:1804.07461 (2018).

[13] A. Chandra, R. Stefanus, Experiments on paraphrase identification using quora question pairs dataset, arXiv preprint arXiv:2006.02648 (2020).

[14] Y. Zhang, J. Baldridge, L. He, PAWS: Paraphrase adversaries from word scrambling, arXiv preprint arXiv:1904.01130 (2019).

[15] J. P. Wahle, T. Ruas, N. Meuschke, B. Gipp, Are neural language models good plagiarists? a benchmark for neural paraphrase detection, 2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL) (2021) 226–229.

[16] M. Eiselt, P. Benno, S. Andreas, A. Barrón-Cedeno, P. Rosso, Overview of the 1st international competition on plagiarism detection, 3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse (2009).

[17] P. Martin, B. Stein, A. Barrón-Cedeno, P. Rosso, An evaluation framework for plagiarism detection, Coling 2010 (2010).

[18] P. Martin, B. Stein, A. Barrón-Cedeno, P. Rosso, PAN Plagiarism Corpus 2011 (PAN-PC-11) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.3250095 (2011).

[19] X. Wei, C. Callison-Burch, W. B. Dolan, Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit), Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015) (2015) 1–11.

[20] W. Lan, S. Qiu, H. He, W. Xu, A continuously growing dataset of sentential paraphrases, arXiv preprint arXiv:1708.00391 (2017).

[21] I. Bensalem, I. Boukhalfa, P. Rosso, L. Abouenour, K. Darwish, S. Chikhi, Overview of the AraPlagDet PAN@ FIRE2015 Shared Task on Arabic Plagiarism Detection, FIRE workshops (2015) 111–122.

[22] N. Babakov, D. Dale, V. Logacheva, A. Panchenko, A large-scale computational study of content preservation measures for text style transfer and paraphrase generation (2022) 300–321.

[23] J. R. Chowdhury, Y. Zhuang, S. Wang, Novelty Controlled Paraphrase Generation with Retrieval Augmented Conditional Prompt Tuning, arXiv preprint arXiv:2202.00535 (2022).

[24] Towards Better Characterization of Paraphrases., author=Liu, Timothy, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2022) 8592–8601.

[25] B. Agarwal, H. Ramampiaro, H. Langseth, M. Ruocco, A deep network model for paraphrase detection in short text messages, Information Processing Management 54 (2018) 922–937.

[26] G. Veena, D. Gupta, L. Amritha, T. A. Athira, Paraphrase detection using deep neural network based word embedding techniques, 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (2020) 517–521.

[27] N. Peinelt, D. Nguyen, M. Liakata, tBERT: Topic models and BERT joining forces for semantic similarity detection, Proceedings of the 58th annual meeting of the association for computational linguistics (2020) 7047–7055.

[28] B. Ko, H.-J. Choi, Paraphrase bidirectional transformer with multi-task learning., 2020 IEEE International Conference on Big Data and Smart Computing (BigComp) (2020) 217–220.

[29] Q. Peng, D. Weir, J. Weeds, Y. Chai, Predicate-Argument Based Bi-Encoder for Paraphrase Identification, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2022) 5579–5589.

[30] J.-M. Torres-Moreno, G. Sierra, P. Peinl, A German Corpus for Text Similarity Detection Tasks, arXiv preprint arXiv:1703.03923 (2017).