GAN-BERT, an Adversarial Learning Architecture for Paraphrase Identification

Hoang Thang Ta[1,2], Abu Bakar Siddiqur Rahman[1,3,*], 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]*College of Information Science and Technology, University of Nebraska Omaha, Omaha, Nebraska, USA*

Abstract

In this paper, we address the task of Paraphrase Identification in Mexican Spanish (PAR-MEX) at sentence-level. We introduced our method, using text embeddings from pre-trained transformer models for the training process by GAN-BERT, an adversarial learning. We modified noises for the generator, which have a random rate and the same size of the hidden layer of transformers. To improve the model performance, a rule of thumb based on the pair similarity is used to remove possible wrong sentence pairs in positive examples; parallel with the addition of unlabelled data in the same domain. The best obtained F1 is 90.22%, ranked third in the final result table, also outperformed the organizers' baseline.

Keywords

Paraphrase Identification, GAN-BERT, Text Classification, PAR-MEX, IberLEF

1. Introduction

Paraphrase is an alternative expression of how to write a text in a same meaning with the original text. Due to high volume of texts in social media, where users tend to post repeatedly the same meaning of texts, there needs to find out the core discussion by automatic summarization approaches. Search engines such as Google and Yahoo are required to identify the paraphrase of words due to different keywords with the same meaning, posted by different users in information searching. For example, a user can search `"Computer Microprocessor"`, while another one with `"Intel"`. Then, the search engines should retrieve nearly the same results for both cases. Similarly, it is needed to identify the lexically or semantically similar texts for question answering systems such as Quora or StackOverflow [1], to filter out the repetitive questions or answers. Lexical similarity means to match by characters or words, whereas semantic similarity refers to the similarity by meaning of two sentences irrespective of words or character matching. One of the problematic parts in academic life is to detect the plagiarism from the published scholarly

IberLEF 2022, September 2022, A Coruña, Spain.

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

articles or the submitted assignments from students in a course. In the United States (U.S.), the office of research integrity has assisted the universities in plagiarism identification [2, 3].

Nowadays, machine learning and deep learning methods have been used recently to detect plagiarism. A convolutional neural network (CNN) and a long short-term memory network (LSTM) with the combination of fine-grained similarity matching models for words are usually applied to detect text paraphrase. Word embeddings use CNN input to learn the local features of words, and then the encoded features from CNN output are fed to LSTM to extract the long term dependencies from the texts. [4]. Hambi et al. created an online plagiarism detection system by using Doc2Vec, Siamese Long Short-term Memory (SLSTM) and CNN with three layers (embedding layer, learning layer and detection layer) [5]. Instead of time consuming for word searching and matching approaches, BERT models performed better in detecting plagiarism, compared to RoBERTa and Glove [6]. Another revolutionary method, transformers in NLP are able to comprehend the semantics of texts to detect plagiarism [7]. Learning the similarity between sentences is one of the important step to detect paraphrase. String similarity measure and maximum entropy classifier is used to recognize paraphrase from a pair of sentences [8].

In this paper, we used GAN BERT in a generative adversarial setting for the task of Paraphrase Identification in Mexican Spanish (PAR-MEX), a competition of IberLEF 2022 [9]. GAN-BERT is a proper practice for cases which there is not enough annotated datasets due to expensive cost and time consuming. Besides, we also introduced a rule of thumb to filter false sentence pairs in positive cases, as well as combine with the available unlabeled data to improve the model performance. As far as we know, our paper is likely the first work to detect paraphrase by using GAN-BERT. Our team finished in third position in the competition (https://codalab.lisn.upsaclay.fr/competitions/2345#results).

2. Related Works

Plagiarism detection is one of the arduous tasks in NLP, as texts contain misspelling, short forms of words or not cleaned [10]. Unlike the other tasks, it is difficult to find a sentence pair in natural texts for building an efficient dataset in paraphrase detection. One of the available datasets is Microsoft Research Paraphrase corpus (MRPC) that consists of 5801 sentence pairs. Heuristic extraction techniques with a support vector machine classifier was applied to select the possible similar sentence pairs from a large corpus of news topics [11]. Human judges assured that 67% of the MRPC dataset are semantically equivalent. Another dataset was collected from Quora Question answering system. Quora Question pairs (QQA) contains question pairs with the same meaning, which are posted by users [12]. Chandra et. al. used this dataset to detect paraphrase from sentence pairs [13]. Later, PAWS (Paraphrase Adversaries from Word Scrambling) was created with more common lexical overlap that consists of 108463 paraphrase and non paraphrase sentence pairs in total. Word swapping and back translation was adopted for challenging sentence pairs, evaluated by human before adding into the dataset [14]. Often, people share the same news on social media. There is a dataset with the tweets that shared in the same news articles. [15]. In 2009, PAN organized a plagiarism competition and released PAN-PC-09 [16]. This dataset was collected from project Gutenberg and includes 41223 text documents [17, 18].

BERT is popular when providing many easy-to-use pre-trained models, can process a high volume of text data and fine-tune to any specific language context. BERT, Latent Dirichlet Allocation (LDA) and Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM) were used in topic modelling to achieve a better performance for paraphrase detection, while GSDMM is suitable for short texts [19]. Paraphrase identification can perform better if it learns sequentially. Ko and Choi proposed Paraphrase-BERT [20], where the pre-trained BERT was fine tuned on MRPC dataset and add a Whole Word Masking. Then, they used multi task learning to improve the performance for identifying paraphrase. Malajyan et. al generated a sentential paraphrase corpus on Armenian language. Initially, the dataset is translated from Armenian to English, and then do back the same thing twice with keeping the same semantic meaning. The dataset has 2360 paraphrases that used BERT to detect paraphrase in Armenian [21]. There is also a work that use BERT to solve the problem of paraphrase detection on Chinese datasets, with results outperformed the baseline [22]. Language-specific BERT was used on different subsets of Opusparcus training data that contain data in English, Finnish, French, German, Russian, and Swedish [23]. In this paper, we used GAN-BERT just to see how well it can for the problem of paraphrase identification. Furthermore, no current work with GAN-BERT on the task, according to our best knowledge, convinced us to try it once.

3. Task Description

Given 2 sentences, the task is to identify whether they have a paraphrase relation or not, and put them correctly into paraphrase (P) or non-paraphrase (NP) categories. From more than 200 Spanish texts in 7 topics (molecular cuisine, sushi, tequila, kebab, vegan food, food truck, and Mexican ofrenda), organizers composed manually sentence pairs, using 3 paraphrase methods, including low paraphrase, high paraphrase, and no paraphrase.

These pairs were modified in difference levels, from word-level by using synonyms or word orders, to sentence-level with sentence structure, just to improve the data variety. Another technique of paraphrase is to keep only the main idea of the source sentence. Besides, a set of common words in 2 texts, having no relation, can be used for the paraphrase.

The challenge has 2 paraphrase methods, element replacement and structure modification, performed on 2 types of paraphrase, high and low-level paraphrases from the only lexical overlap of the sentences. Paraphrase Identification in Mexican Spanish (PAR-MEX) is the first edition for the task of paraphrase identification in Spanish, and its paraphrase process follows the method of Torres-Moreno et al., 2014 in constructing a German corpus [24].

4. Dataset Analysis

There are 3 datasets, training set, validation set, and test set offered by organizers from GitHub (https://github.com/GIL-UNAM/PARMEX_2022) or the challenge website (https://-codalab.lisn.upsaclay.fr/competitions/2345). Figure 1 displays some basic statistics of these datasets. The training set includes 7,382 sentence pairs, while 97 and 2,891 are the number of pairs in 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. In detail, P pairs are

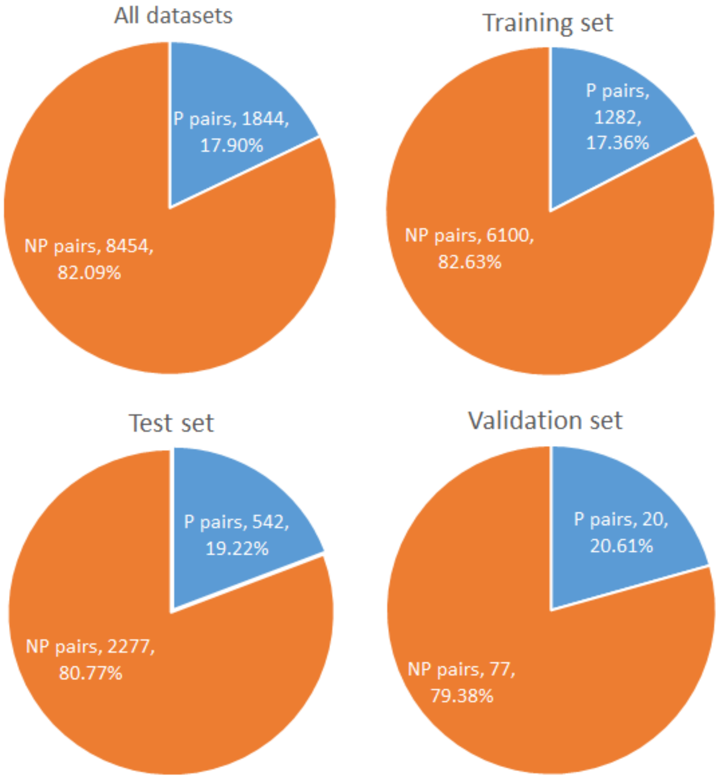

Figure 1: The distribution of categories in various datasets.

accounted for 17.35% in the training while NP pairs are 82.63%. Similarity, in the test set, the distribution of P pairs and NP pair is 17.90% and 82.09%.

Next, we perform the distribution of length (number of tokens) by text in the training set. For each pair, 2 sentences will be concatenated by a space and put its result to spaCy v2.3.2 (https://spacy.io/) for parsing and count its length. Note that this practice is just only to see the distribution of length by text, shown in Figure 2. For the creation of training data, we need to add more 2 special tokens, `[CLS]` at the beginning, and `[SEP]` in between of 2 sentences and at the end. The maximum length is 135, so we can set hyperparameter `"MAX_LEN=160"` in the training to guarantee that the model can capture all tokens in the token encoding. However, most text lengths fall in the range `[20, 80]` so one can consider to use this value for the training with a proper text truncation.

5. Methodology

There is no need to have any preprocessing step when all datasets seems to be cleaned, according to our observations. However, we still suggest to use package `es_core_news_md` of spaCy to remove redundant spaces and special symbols, or normalize several punctuations if possible.

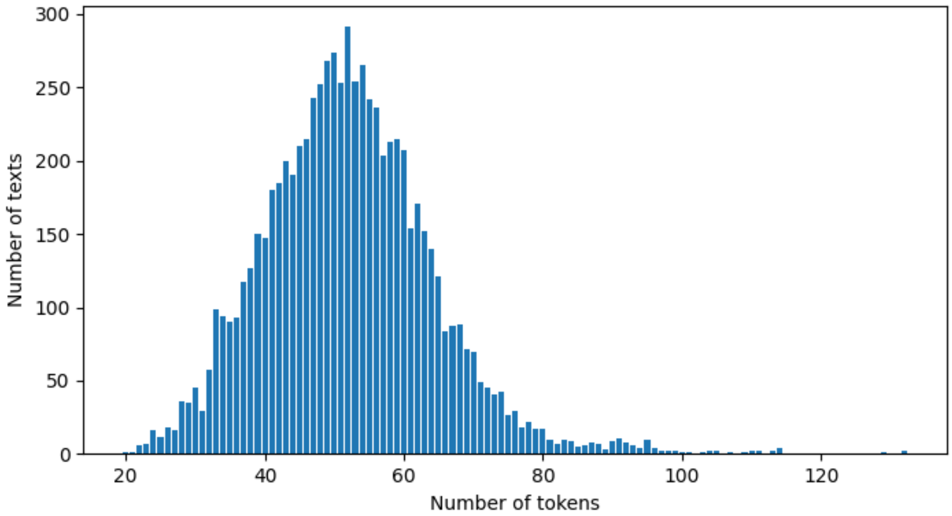

Figure 2: The distribution of categories in various datasets.

Stopwords will not be removed because there is a case related to the problem of negation, which leads to false paraphrase pairs [25]. Such word `"not"` or its short form `"n't"` may contribute a negative meaning in sentences. For example, `"I don't like kebab. (No me gusta el kebab.)"` is not a paraphrase of `"I like kebab. (Me gusta el kebab.)"`. But they will a paraphrase pair if we remove word `don't` in English (or `no` in Spanish). To best of our scanning, we did not see any false positive cases in the datasets. However, it is an interesting point if the organizers add this problem in the future challenge.

When working with the datasets, we found noises as false positive or true negative sentence pairs. This may be intentionally from organizers to increase the task difficulty or unintentionally in data creation which can not control well the generated examples. For quick, we decided to use a rule of thumb, based on the pair similarity from a pre-trained model, `"symanto/sn-xlm-roberta-basesnli-mnli-anli-xnl"` in Hugging Face. This RoBERTa model can support up to 514 tokens, which allow us to search for the similarity between 2 longer sentences. For pair similarity of the 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 this paper, we apply GAN-BERT [26] for the data training, a BERT-like architecture with unlabeled data in a generative adversarial setting. GAN-BERT is proper for small datasets which could not be collected more data because this task is expensive and time consuming. Figure 3 describes the architecture of GAN-BERT. Traditionally, SS-GAN include 2 networks: a discriminator D for classifying input dataset, (2) a generator G to differentiate fake data in an adversarial manner. The original authors added BERT on the top of SS-GAN to provide sentence embeddings and contextualized embeddings of the words in a sentence.

The real data consists of 2 categories (labeled and unlabeled) will be passed to BERT to get

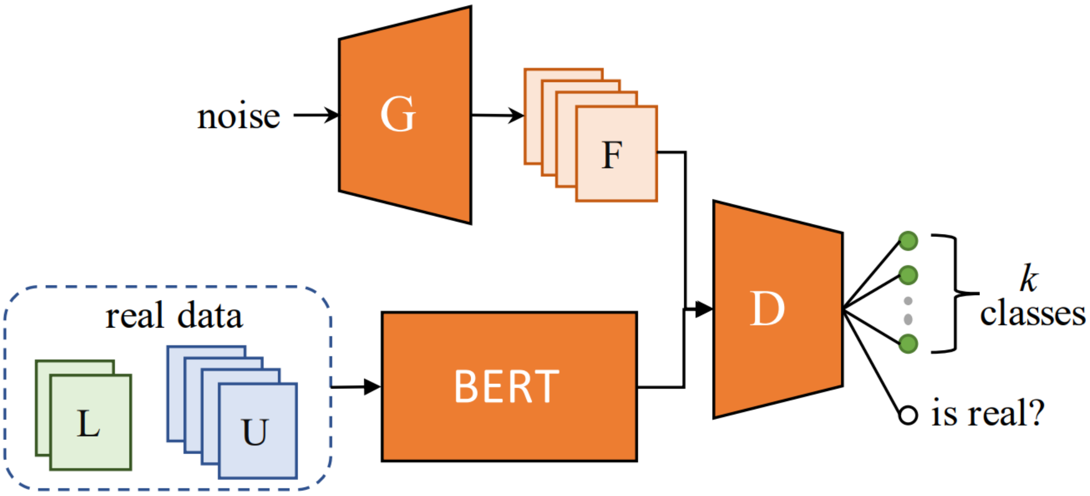

Figure 3: GAN-BERT architecture: A set of fake sentence pairs F, given a random distribution, will be generated by G. The discriminator D will take these fake pairs, unlabeled U and labeled L vector representations computed by BERT as input for the training process [26].

text embeddings. For example, the training set is labelled data because its sentence pairs was already labeled as 0 (NP, non-paraphrase) or 1 (P, paraphrase). The test set and the validation are unlabelled data because their data is unknown. In this paper, because there have only 2 categories (P and NP), so we set P pairs as labeled data, otherwise data (include NP pairs) is unlabeled. Meanwhile, noises or fake sentence pairs will be generated based on real text embeddings (vectors). The original paper generated noises as 100-dimensional vectors drawn from a normal distribution *N(0, 1)*. However, we did a different way. When real examples generated from BERT, we took their hidden layer embeddings (768-dimensional), generated a noise rate, then distorted the real data by this rate with normal distribution *N(-1, 1)*. This practice helps to have noises closer to the real data. Next, the discriminator D will learn and differentiate between real and fake sentence pairs. If a sentence pair is real, it will be classified to one of k classes. In contrast, this pair will be classified to class $k + 1$ if it is fake.

6. Experiment

We add `[CLS]` and `[SEP]` tokens to sentence pairs to build input data for GAN-BERT, in the form of `"[CLS] sentence 1 [SEP] sentence 2 [SEP]"`. Our GAN-BERT contains only 2 categories, P and UNLABELED. Sentence pairs of P category is used because they take the minority part in the training set, likely help the discriminator learn faster and control the data easier. Other pairs, in NP category and in other sets, will be put in UNLABELED category. If there are available unlabeled data in the same domain, we can add them to improve the amount of unlabeled data. The more unlabeled data helps the model benefit more from the adversarial learning [27] and also improve the model's inner representation [26].

We use a pre-trained model to provide sentence embeddings for the discrimina-

Table 1
Our submissions' results from the organizer website.

#	Method	Noise filtering	F1*	Unlabeled data
1	GANBERT	No	0.8609	test_set
2	**GANBERT**	**Yes**	**0.9022**	test_set + val_set
3	GANBERT	Yes	0.8438	none
4	*Baseline (BETO & BERT)*	-	*0.7026*	-

*The results taken from the organizers' website.

tor. Some main hyperparameters for the whole architecture of GAN-BERT are `"max_seq_length = 160"`, `"seed_val = 42"`, `"learning_rate_discriminator = 2e-7"`, `"learning_rate_generator = 2e-7"`, `"num_train_epochs = 20"`, and `"batch_size = 16"`. Although there is a limitation of computer resources in our server, a larger batch size can be deployed because the discriminator is a multi layer perceptron network, not require so much memory in the training. We used the same and small learning rates for discriminator and generator. Discriminator's learning rate is more important when it helps the model reach the higher accuracy. All hyperparameters are also the same for the training and prediction phrases to guarantee the model stability in the prediction. Here is a list of other hyperparameters:

- `"num_hidden_layers_g = 2"`: This is the number of hidden layers of the discriminator.
- `"num_hidden_layers_d = 2"`: This is the number of hidden layers of the generator.
- `"noise_size = 768"`: As mentioned, the noise will be generated with the same size as the hidden size of the transformer. The rate of noises is randomly, in range [0,1]. Noises themselves take value from a normal distribution *N(-1,1)*.
- `"apply_balance = False"`: Apply only if the labeled data contains less than `1%` of datasets.
- `"epsilon = 2e-7"`: This value is added to loss functions to avoid the case of zero logarithm in calculations.

We set up 3 runs with different configurations, shown in Table 1. The first run had no noise filtering and test set was added as unlabeled data. The second run had noise filtering, using a rule of thumb (remove P pairs < 0.3454 (Avg. All) and add NP pairs > 0.7978 (Avg. P) to the training set). The unlabeled data of this run was test and validation sets. The last run had no unlabeled data, but applied noise filtering (remove P pairs < 0.45 (Avg. All) and add NP pairs > 0.7978 (Avg. P) to the training set). We trained on the whole training set, and saved the best model with the highest accuracy after 20 epochs. The accuracy is an average value of training accuracy and validation accuracy. Since validation set contains only `100` sentence pairs, we manually labeled them with our best knowledge (not Spanish natives) and obtained the best score of 95.24%. At last, we obtained the best F1 of 90.22% in the second run with noise filtering and the most amount of unlabeled data. Our best F1 is ranked third in the final result table, provided by organizers.

There are some knowledge that we have learned in our experiments. The lower learning rate helps model learn better, but not to set this value too low. If so, the accuracy can not increase or increases slowly after every epoch. Noise filtering is an important sub-task for paraphrase identification by GAN-BERT. We only need to filter noises with categories in the discriminator. In our case, to filter noises in P pairs is enough, the model will learn positive examples and push the negative (or fake) ones out. There is a limit for the training accuracy, depending on the percentage of unknown positive examples in the addition unlabeled data, not count noises as false sentence pairs.

7. Conclusion

In this paper, we apply GAN-BERT, an adversarial learning, which trained input data as text embeddings extracted a pre-trained model for paraphrase identification in Mexican Spanish at a sentence-level. Different from the original architecture of GAN-BERT, we modify noises fed the generator with a random rate are the same size as the hidden layer of transformers. A rule of thumb based on sentence pair similarity is used to filter false sentence pairs in positive examples. We executed our method in different runs, combining with/without noise filtering and/or the addition of unlabeled data. The best obtained F1 was 90.22%, which outperformed the baseline offered by organizers, and ranked third in the final result table.

In the future, we will investigate deeply about the correlations between the amount of unlabeled data and model performance in GAN-BERT, which was not so clear just by some experiments in this paper. Furthermore, we also concern to the effective noise filtering methods and try GAN-BERT on other datasets such as MRPC or QQP. At last, we will try with other architectures of adversarial learning or modifications of GAN-BERT to look for a better solution in paraphrase identification.

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, University of Nebraska to provide their high computing GPU resources. The authors acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

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

[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] 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.

[5] E. M. Hambi, F. Benabbou, A new online plagiarism detection system based on deep learning, International Journal of Advanced Computer Sciences and Applications 11 (2020) 470–478.

[6] R. Rosu, A. S. Stoica, P. S. Popescu, M. C. Mihăescu, NLP based Deep Learning Approach for Plagiarism Detection, RoCHI-International Conference on Human-Computer Interaction, Romania (2021).

[7] C. Amzuloiu, M. C. Mihăescu, T. Rebedea, Combining Encoplot and NLP Based Deep Learning for Plagiarism Detection, International Conference on Intelligent Data Engineering and Automated Learning, Springer, Cham (2021) 97–106.

[8] P. Malakasiotis, Paraphrase recognition using machine learning to combine similarity measures, Proceedings of the ACL-IJCNLP 2009 Student Research Workshop (2009) 27–35.

[9] 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).

[10] 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.

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

[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] 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.

[20] 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.

[21] A. Malajyan, K. Avetisyan, T. Ghukasyan, ARPA: Armenian Paraphrase Detection Corpus and Models, 2020 Ivannikov Memorial Workshop (IVMEM), IEEE (2020) 35–39.

[22] B. An, Chinese Paraphrase Dataset and Detection, International Conference on Asian

Language Processing (IALP) (2021) 235–240.

[23] T. Vahtola, M. Creutz, E. Sjöblom, S. Itkonen, Coping with Noisy Training Data Labels in Paraphrase Detection, Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021) (2021) 291–296.

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

[25] M. Mohamed, M. Oussalah, A hybrid approach for paraphrase identification based on knowledge-enriched semantic heuristics, Language Resources and Evaluation 54 (2020) 457–485.

[26] D. Croce, G. Castellucci, R. Basili, GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples, in: Proceedings of the 58th annual meeting of the association for computational linguistics, 2020, pp. 2114–2119.

[27] C. Breazzano, D. Croce, R. Basili, MT-GAN-BERT: Multi-Task and Generative Adversarial Learning for sustainable Language Processing (2021).