International Journal of User-System Interaction 13(1) 2020, 48-60 © MatrixRom

NLP based Deep Learning Approach for Plagiarism Detection

Razvan Rosu, Alexandru Stefan Stoica, Paul Stefan Popescu, Marian Cristian Mihăescu
University of Craiova
Department of Computers and Information Technology

{Celtics.razvan,stoicaastefan}@gmail.com, {spopescu, mihaescu}@software.ucv.ro

Abstract. Plagiarism detection represents an application domain for the NLP research area, which has not been investigated too much by researchers in the context of lately developed attention mechanism and sentence transformers. In this paper, we present a plagiarism detection approach which uses state-of-the-art deep learning techniques in order to provide more accurate results than classical plagiarism detection techniques. This approach goes beyond classical word searching and matching, which is time-consuming and can be easily cheated because it uses attention mechanisms and aims for text encoding and contextualization. In order to get proper insight regarding the system, we investigate three approaches in order to be sure that the results are relevant and well-validated. The experimental results show that the systems that use BERT pre-trained model offers the best results and outperforms GloVe and RoBERTa

Keywords: BERT, RoBERTa, sentence transformers, plagiarism, NLP

DOI: 10.37789/ijusi.2020.13.1.4

1. Introduction

Text similarities and plagiarism detection is a well-known issue in natural language processing (NLP) research area. One of the most critical challenges in this area is to optimize the results and to reduce the time spent on document analysis. There are several approaches used for plagiarism detection, most of them implying text parsing using different algorithms and setting thresholds for the number of words that matches.

In this paper, we present an innovative approach which relies on deep learning techniques along with some basic techniques which allow us to benchmark our results. This approach uses sentence transformers Ashish et.al. (2017) which uses pre-trained models from different neural networks and on different datasets. This approach is entirely new, and the main difference comparing to classical deep learning approaches is that in the transformers, every time-step has access to all previous time-steps Merkx et. al. (2020) and makes use of attention mechanisms Hu (2019). Another benefit of using this approach is that despite that deep learning models take time to train and tune in order to get the best results, they are fast when using for obtaining results.

In our approach, we have used BERT Liu et. al. (2019), which is a bidirectional encoder representation from transformers used mainly for understanding the user searches from Google search engine. The motivation for choosing this approach is that since BERT was launched, it obtained an adequate level of performance for relevant tasks like GLUE (General Language Understanding Evaluation), SquAD (Stanford Question Answering Dataset) or SWAG (Situations with Adversarial Generations). The other approach we considered is GLoVE Pennington (2014) combined with TF-IDF Ramos (2003) and cosine similarity Thada (2013) because GLoVE can accomplish easily two goals: to create word embeddings in space vectoring and considers global rather than local statistics.

The novelty of our approach is that BERT and RoBERTa were not used before for plagiarism detection and we aim to investigate if using this approach may lead us to a better and more relevant ranking of the documents that are similar to the query document.

We aim to produce a robust plagiarism detection mechanism which can provide both accurate and fast enough results. In most of the cases, plagiarism detection means the word by word comparison and which takes plenty of time to parse a large document and can be easily tricked by changing words with their synonyms. Using deep learning which aims for understanding the text rather than just parsing words can reveal a new level of plagiarism detection.

2. Related Works

Plagiarism detection using different algorithms started a long time ago Parker & Hamblen (1989). However, it is still an actual problem as the amount of data and the complexity of the techniques used for avoiding the detection mechanisms raised. Starting from whose five elves of plagiarism defined in Parker & Hamblen (1989) and going to the actual approach of deep learning

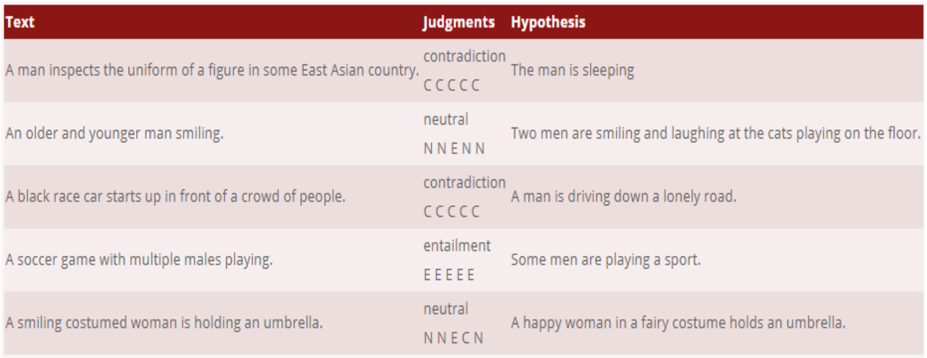
Bakhteev et. al. (2019) a significant amount of work has been published, but there are still open problems Ko & Choi (2020)

Our approach of using BERT for plagiarism detection is entirely new, and there are several papers which prove its efficiency. For example, in Lukashenko et. al. (2007), the authors propose a paraphrase-BERT to perform a task for paraphrase identification. The authors first fine-tune the pre-trained BERT with MRPC dataset, and then, they add a Whole Word Masking, which is pretraining method recently published by Google for BERT. In the end, they perform Multi-Task Learning (MLT) to improve performance. More precisely, the Question Answering task and the Paraphrase Identification task are learned sequentially to improve performance of Paraphrase Identification. As a result, the authors show that MLT affects a performance improvement of downstream task by 11.11%-point absolute accuracy improvement and 7.88%- point absolute F1 improvement.

Another recently published paper Huang et. al. (2020) describes TRANS-BLSTM, which comes from the Transformer with Bidirectional LSTM for Language Understanding. In paper Huang et. al. (2020), the authors investigate how these two modelling techniques can be combined to create a more robust model architecture. They propose a new architecture which combines as a Transformer with BLSTM (TRANS-BLSTM) which has a BLSTM layer integrated for each transformer block, leading to a combined modelling framework for transformer and BLSTM. The authors present that TRANS-BLSTM models consistently lead to improvements in accuracy compared to simple BERT baselines in experiments on GLUE and SQuAD 1.1. Their TRANS-BLSTM model obtains an F1 score of 94.01% on the SQuAD 1.1 development dataset, which is comparable to the state-of-the-art result.

Another recent paper Iandola et. al. (2020) which addresses the NLP problems using BERT presents SqueezeBERT and analyzes what can computer vision teach NLP about efficient neural networks. The authors address the computational performance problems as today's highly accurate NLP deep learning models such as BERT and RoBERTa are extremely computationally expensive, with BERT-base taking 1.7 seconds to classify a text snippet on a Pixel 3 smartphone. Their novel network architecture called SqueezeBERT runs 4.3x faster than regular BERT on the Pixel 3 while achieving competitive accuracy on the GLUE test set.

Regarding BERT, it has been previously used with success for question answering as stated in Pîrtoacă et. al. (2019) and to the best of our knowledge BERT has been used for plagiarism detection only in Zubarev & Sochenkov (2019) . Based on the results obtained on paper Pîrtoacă et. al. (2019) we have been inspired to use this approach in a different context because at this moment plagiarism detection is a high interest subject. Regarding the results comparison we use Cosine similarity and a fully accurate explanation can be found on Machine Learning Mastery example which presents three documents from which authors extracted three common words.



Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

Figure 1. SNLI dataset sample

3. System design and implementation

In this paper, we considered two training datasets: SNLI Bowman et. al. (2015) for BERT and MultiNLI Williams et. al. (2017) for TF-IDF with GLoVE. Regarding SNLI dataset, a short sample is presented in Figure 1 and contains 570.000 of pairs of sentences, human written in English. MultiNLI (Multi-Genre Natural Language Inference) have a similar structure to SNLI having 433000 pairs of sentences but also differs from having a series of text genders. The models based on these datasets were gathered from an external library because the training process would have taken too long. The models were trained using STS benchmark dataset O'shea et. al. (2017), which has a selection of English datasets which were organized in the context of SemEval between 2012 and 2017.

The architecture of a transformer contains two main components: the encoder and the decoder. The encoder contains a layer called Multi-Head Attention, followed by another layer called Feed Forward Neural Network. The decoder contains the previously mentioned layers along with a layer called Masked Multi-Head Attention.

Figure 2 presents the application's data analysis workflow. Our approach starts from pre-trained models for Glove, BERT and RoBERTa, and we first apply fine-tuning using STS benchmark in order to obtain relevant results for plagiarism detection. Based on the models we obtained after the fine-tuning step and the documents used for query and index, we create the word embeddings which can be used for further ranking and model evaluation.

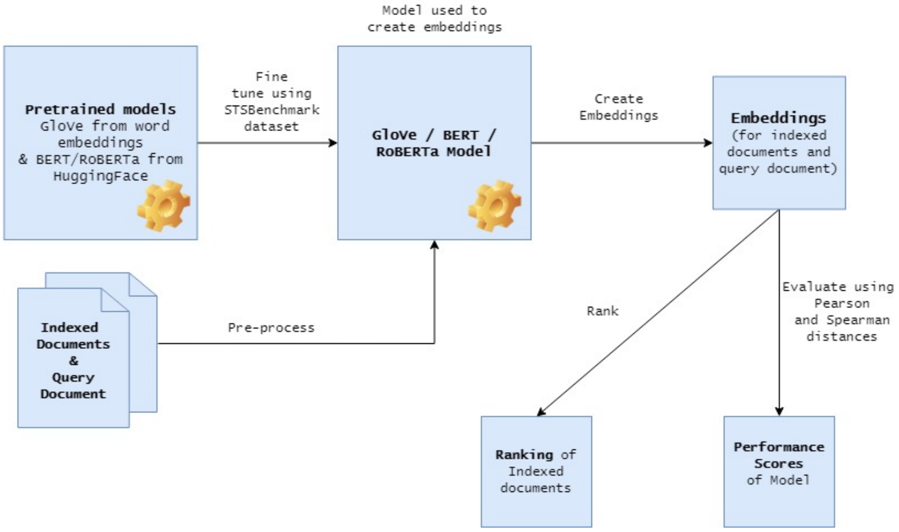

Model used to create embeddings
Pretrained models Glove from word embeddings & BERT/RoBERTa from HuggingFace
Fine tune using STSBenchmark dataset
Glove / BERT / RoBERTa Model
Create Embeddings
Embeddings (for indexed documents and query document)
Indexed Documents & Query Document
Pre-process
Rank
Evaluate using Pearson and Spearman distances
Ranking of Indexed documents
Performance Scores of Model

Figure 2. Data Analysis Workflow

The BERT Structure, which uses the above-mentioned transformers use two flows of input and output. The input is a sequence of tokens, which are first embedded into vectors and then processed in the neural network. The output is a sequence of vectors of size H, in which each vector corresponds to an input token with the same index. Regarding BERT, one thing that needs to be mentioned is that before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence. In technical terms, the prediction of the output words requires:

1. Adding a classification layer on top of the encoder output.
2. Multiplying the output vectors by the embedding matrix, transforming

them into the vocabulary dimension.

3. Calculating the probability of each word in the vocabulary with SoftMax.

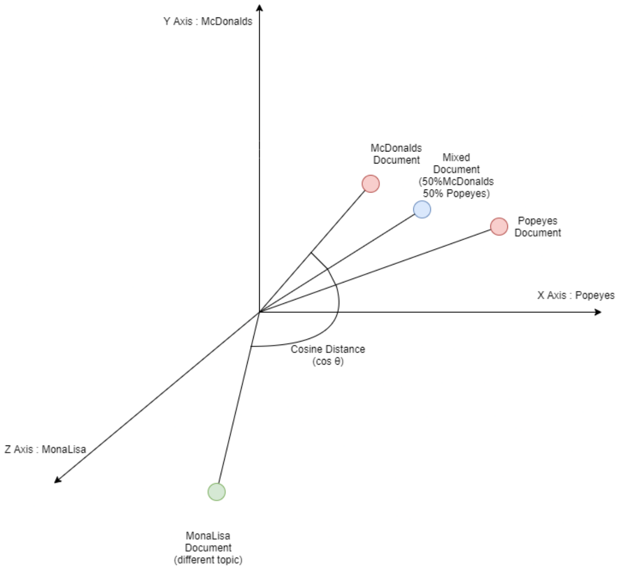


Figure 3. Sample cosine similarity results

Figure 3 presents the cosine similarity computed for the four documents used in our paper. In order to better exemplify the mechanism, we consider three common documents: McDonalds, Popeys and a mix of these two along with a different topic document (i.e., MonaLisa). Projecting this document into a 3D space, the first two documents are close to each other because they share a specific subject and the last one refers to a different subject. Even if by parsing words we can get a similar doc and a short Cosine Distance, using BERT will provide a result similar to the one exemplified in Figure 3 which reveals a long distance between the first three documents and the one with MonaLisa.

4. Experimental Results

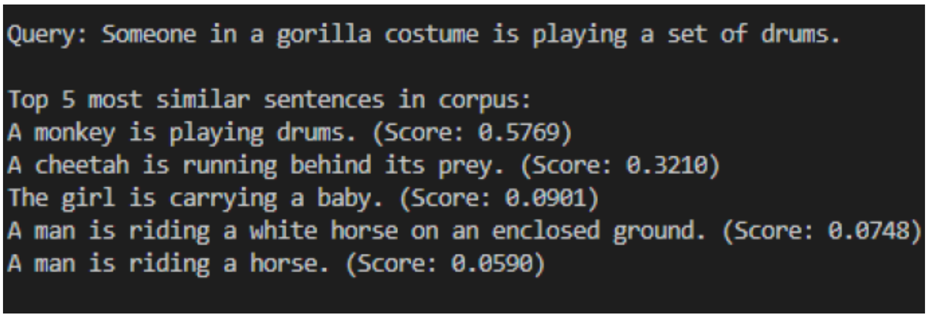
The experiments are conducted in three main directions, one on Glove and

TF-IDF, the second which uses BERT and the final one which uses RoBERTa. Each of these approaches was evaluated in two ways, one on the training dataset and one on completely unseen data which allows us to further evaluate the models. For experiments replication, the code and data used are available on GitHub

Table 1. Results for word embeddings using GloVe+TF-IDF

Epoch and corr.	**Cosine Similarity**	**Manhattan distance**	**Euclidean Distance**	**Dot Product Similarity**
0 P	0.7590	0.7138	0.7153	0.5730
0 S	0.7554	0.7250	0.7271	0.6186
1 P	0.7738	0.7151	0.7165	0.5971
1 S	0.7713	0.7272	0.7290	0.6522
2 P	0.7777	0.7153	0.7167	0.5969
2 S	0.7746	0.7281	0.7296	0.6559
3 P	0.7791	0.7154	0.7168	0.5963
3 S	0.7762	0.7285	0.7300	0.6561
4 P	0.7801	0.7154	0.7168	0.5961
4 S	0.7770	0.7287	0.7301	0.6564

Table 1 presents the results obtained using GloVe and TF-IDF techniques having a batch size of 32 words over five epochs, and the evaluation process was made on Wikipedia Document Frequencies dataset . On the first column, there is the epoch number, and we also present which correlation is used; P stands for Pearson and S for Spearman correlation. On the next columns, we have the metrics used for each correlation computation, and the small number of epochs is motivated by the almost no increase in performance among the last epochs we trained. We use four metrics: Cosine similarity, Manhattan distance, Euclidean Distance and Dot Product Similarity because we need to be sure that the results are relevant and unbiased.



```
Query: Someone in a gorilla costume is playing a set of drums.

Top 5 most similar sentences in corpus:
A monkey is playing drums. (Score: 0.5769)
A cheetah is running behind its prey. (Score: 0.3210)
The girl is carrying a baby. (Score: 0.0901)
A man is riding a white horse on an enclosed ground. (Score: 0.0748)
A man is riding a horse. (Score: 0.0590)
```

Figure 4. Validation example for GloVe+TF-IDF

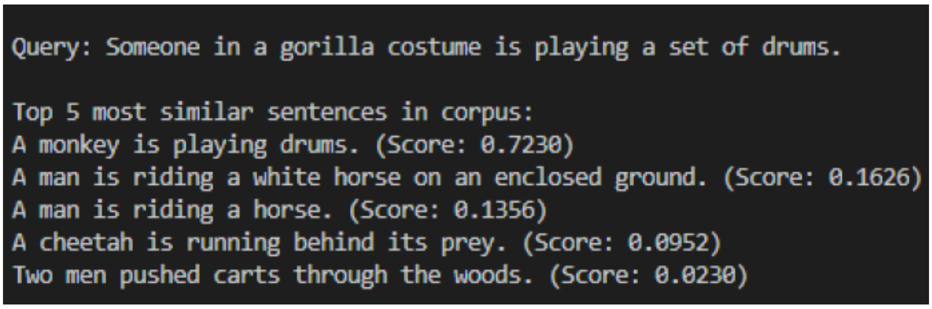
Figure 4 presents a validation example for the Glove and TF-IDF approach. The query is specified on the first row, and after that, we have the top five most similar sentences from the corpus along with the matching score. We use this approach as a baseline for BERT and RoBERTa frameworks.

Further, the BERT model was trained over four epochs as it is a pretrained model. This helps the model to be fine-tuned for plagiarism detection.

Table 2. BERT training results

Epoch and corr.	Cosine Similarity	Manhattan distance	Euclidean Distance	Dot Product Similarity
0 P 0 S	0.8685 0.8721	0.8410 0.8490	0.8407 0.8487	0.8069 0.8121
1 P 1 S	0.8751 0.8759	0.8442 0.8499	0.8436 0.8493	0.8155 0.8211
2 P 2 S	0.8765 0.8773	0.8510 0.8551	0.8501 0.8545	0.8275 0.8313
3 P 3 S	0.8767 0.8775	0.8519 0.8562	0.8511 0.8556	0.8268 0.8307

Table 2 presents the results obtained after training BERT only for just four epochs, and we can easily see that there are better results than in the previous approach. The motivation for stopping at only four epochs trained is that the model performs well at our tests, and there is no significant increase in performance between last epochs.



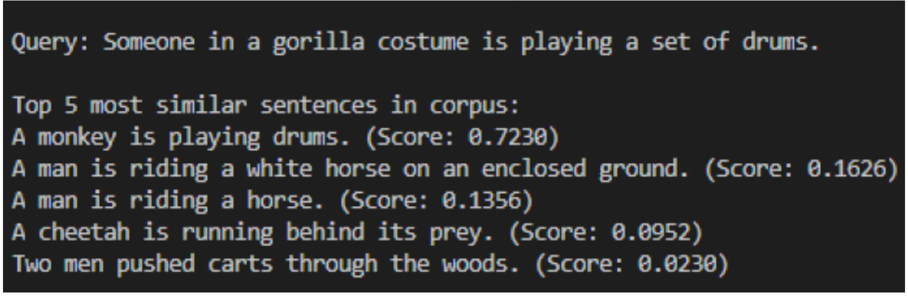
```
Query: Someone in a gorilla costume is playing a set of drums.

Top 5 most similar sentences in corpus:
A monkey is playing drums. (Score: 0.7230)
A man is riding a white horse on an enclosed ground. (Score: 0.1626)
A man is riding a horse. (Score: 0.1356)
A cheetah is running behind its prey. (Score: 0.0952)
Two men pushed carts through the woods. (Score: 0.0230)
```

Figure 5. Validation example for BERT

Figure 5 presents the validation example for BERT approach using the same set of sentences. When comparing the scores, we can see that the similarity distribution is changed and having a better score for the first sentence and less for the others which are not relevant for the query. This reveals a better result as "A monkey is playing a drum" is closer in terms of meaning to the query question.

For training RoBERTa we also used four epochs and got similar results for both the approaches as we can see in Figure 6.



```
Query: Someone in a gorilla costume is playing a set of drums.

Top 5 most similar sentences in corpus:
A monkey is playing drums. (Score: 0.7230)
A man is riding a white horse on an enclosed ground. (Score: 0.1626)
A man is riding a horse. (Score: 0.1356)
A cheetah is running behind its prey. (Score: 0.0952)
Two men pushed carts through the woods. (Score: 0.0230)
```

Figure 6. Validation results for RoBERTa

We also perform validation on several documents, one used as query and the others with different similarity levels. Our approach was to compare the document used as query iteratively with the rest of the documents, including itself and to compute the similarity percentage. There are eleven documents which had medium length containing 6-7 paragraphs, each of them 4 to 5 sentences long. The documents were chosen in order to highlight several scenarios like perfect matching, good matching, 50% matching, low similarity and no similarity.

A short comparison of Figure 4,5 and 6 reveals that BERT and RoBERTa are able to offer a better ranking on the sentence which refers monkeys (0.76 and 0.72) than the standard GLOVE+TF-IDF which reveals a score of just 0.57.

Table 3. Indexed documents

ID	**Sample Text**	**Info about text** **[Topic]: [about]**
text1a	"Popeyes is an American multinational chain of fried chicken …"	Fast Food: Popeyes
text1b	"McDonalds Corporation is an American fast food company …"	Fast Food: McDonalds
text1ab	"Since 2008, it's full brand name is Popeyes…	Fast Food: 50% from text1a, 50% from text1b
text2a	"A plum is a fruit of the subgenus…"	Fruits: Plums
text2b	"A cherry is the fruit of many plants of the genus prunus.."	Fruits: Cherry
text3a	"The Mona Lisa is a half-height portrait painting.."	Art: Mona Lisa
text3b	"The Kiss is an oil-on-canvas painting with added gold.."	Art: The Kiss
text4a	"Computer Science is the study of computation.."	Science: Computer Science
text4b	"Machine learning is the study of computer algorithms…"	Science: Machine Learning
text5a	League of Legends is a multiplayer online battle arena video game…"	Games: League of Legends
text5b	Dota2 is a multiplayer online battle arena developed…"	Games: Dota2

Table 3 presents a short intuition about the documents used for validation. The complete text of the documents can be found on GitHub. On the first column we have the document's id as it can be found on the github and on the next colum a sample of the text. Last column is reserved for a short intuition regarding the document's area and it's useful for better understanding the ranking presented on Table 4. First three documents (text1a, text1b and text1ab) are related and described fast food chain companies, the next two documents refer fruits which also represents food but without many common words and we choose to have this setup in order to evaluate the results offered by the system. Text1ab documents is a mix of text a and text b, this document is also used for system validation and it should bring a score better than 1b and less than 1a for an ideal system.

Table 4. Rankings on indexed documents

BERT	**GloVe**	**RoBERTa**
docID-Similarity	**docID-Similarity**	**docID-Similarity**
text1a-1.00	**text1a-1.00**	**text1a-1.00**
text1ab-0.83	text1ab-0.88	text1ab-0.83
text1b-0.78	text1b-0.80	text1b-0.79
text2a-0.42	text**5b-0.35**	text5b-0.34
text2b-0.41	text2a-0.35	text3a-0.38
text5b-0.41	text4a-0.25	text5a-0.36
text5a-0.38	text5a-0.22	text4a-0.34
text4a-0.33	text2b-0.21	text3b-0.30
text3a-0.29	text4b-0.18	text2b-0.29
text3b-0.27	text3a-0.16	text4b-0.28
text4b-0.26	text3b-0.16	text2a-0.25

Table 4 presents the results obtained with the proposed methods on the previously mentioned documents. The first row of the table represents the method used for plagiarism detection and on the next row the format of the result. First, we define the document ID as it can be found on the GitHub project and then the similarity score. Documents are divided into a and b parts indicating their relation; for example, doc 1a is related to 1b, 2a to 2b and so on. The document marked as 1a is a document identical to the first one and 1ab is a document composed 50% from 1a and 50% from 1b. As we can see, each of the methods detects very well the plagiarism and also detects if part of the text is matched. One thing that needs to be mentioned is that despite the good score obtained by GloVe method, BERT offers the most relevant results understanding the context and better document ranking relative to the query.

5. Conclusions

This paper proposes a plagiarism detection mechanism which goes beyond regular words matching. This approach is useful because the system is able to understand the content and is able to find plagiarism even when several cheating techniques are applied. The system validation methods reveal good results on both benchmark data and our validation data for all the three approaches, but BERT approach differentiates from the other two by context understanding and offer better ranking results. Another interesting conclusion which needs to be mentioned is that BERT and RoBERTa performs significantly better on short documents as the length of the document grows, the accuracy tends to decrease so there are several methods that need to be investigated in order make the system more generic and robust.

Our technique is new and innovative and as this is a first attempt and the results are particularly good, we aim for further development and performance improvement. One limitation we aim to investigate is to make the system perform well on long texts as longform Beltagy et. al. 2020. The second is to produce a generic solution for a variety of texts which can offer both good and relevant results.

References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aiden N. Gomes, Lukasz Kaiser, and Illia Polosukhin.(2017). *Attention is all you need.* In 31st Conference on Neural Information Processing Systems (NIPS 2017), pages 6000–6010.

Bakhteev O, Ogaltsov A, Khazov A, Safin K, Kuznetsova R. (2019) *CrossLang: the system of cross-lingual plagiarism detection*. InWorkshop on Document Intelligence at NeurIPS 2019 Sep 14.

Beltagy, Iz, Matthew E. Peters, and Arman Cohan. (2020) *Longformer: The long-document transformer.* arXiv preprint arXiv:2004.05150.

Bowman SR, Angeli G, Potts C, Manning CD. (2015) *A large annotated corpus for learning natural language inference.* arXiv preprint arXiv:1508.05326. 2015 Aug 21.

Williams A, Nangia N, Bowman SR. (2017) *A broad-coverage challenge corpus for sentence understanding through inference*. arXiv preprint arXiv:1704.05426. 2017 Apr 18.

Iandola FN, Shaw AE, Krishna R, Keutzer KW. (2020) *SqueezeBERT: What can computer vision teach NLP about efficient neural networks?.* arXiv preprint arXiv:2006.11316. 2020 Jun 19.

Huang Z, Xu P, Liang D, Mishra A, Xiang B. (2020) *TRANS-BLSTM: Transformer with Bidirectional LSTM for Language Understanding*. arXiv preprint arXiv:2003.07000. 2020 Mar 16.

Hu D. (2019) *An introductory survey on attention mechanisms in NLP problems.* In Proceedings of SAI Intelligent Systems Conference 2019 Sep 5 (pp. 432-448). Springer, Cham.

Ko B, Choi HJ.(2020) *Paraphrase Bidirectional Transformer with Multi-task Learning.* In IEEE International Conference on Big Data and Smart Computing (BigComp) 2020 Feb 19 (pp. 217-220). IEEE.

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. (2019*) Roberta: A robustly optimized bert pretraining approach*. arXiv preprint arXiv:1907.11692. 2019 Jul 26.

Lukashenko R, Graudina V, Grundspenkis J. (2007) *Computer-based plagiarism detection methods and tools: an overview*. In Proceedings of the 2007 international conference on Computer systems and technologies 2007 Jun 14 (pp. 1-6).

Merkx D, Frank SL.(2020) *Comparing Transformers and RNNs on predicting human sentence processing data.* arXiv preprint arXiv:2005.09471. 2020 May 19.

O'shea J, Bandar Z, Crockett K. (2014) *A new benchmark dataset with production methodology for short text semantic similarity algorithms*. ACM Transactions on Speech and Language Processing (TSLP). 2014 Jan 3;10(4):1-63.

Pennington J, Socher R, Manning CD. (2014) *Glove: Global vectors for word representation*. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) 2014 Oct (pp. 1532-1543).

Ramos J.(2003) *Using tf-idf to determine word relevance in document queries.* In Proceedings of the first instructional conference on machine learning 2003 Dec 3 (Vol. 242, pp. 133-142).

Parker A, Hamblen JO. (1989) *Computer algorithms for plagiarism detection*. IEEE Transactions on Education. 1989 May;32(2):94-9.

Pîrtoacă, George-Sebastian, Traian Rebedea, and Stefan Ruseti. (2019) *Answering questions by learning to rank--Learning to rank by answering questions*. arXiv preprint arXiv:1909.00596 (2019).

Thada V, Jaglan V.(2013) *Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm*. International Journal of Innovations in Engineering and Technology. 2013 Aug;2(4):202-5.

Zubarev, D. V., & Sochenkov, I. V. (2019). *Cross-language text alignment for plagiarism detection based on contextual and context-free models*. In Proc. of the Annual International Conference "Dialogue (Vol. 1, pp. 799-810)