Incoherent Sentence Detection in Scientific Articles in Russian and English

Quang Huy Nguyen
St. Petersburg Electrotechnical University
Saint Petersburg, Russia
nguyenquanghuy1997@gmail.com

Mark Zaslavskiy
St. Petersburg Electrotechnical University, JetBrains Research
Saint Petersburg, Russia
mark.zaslavskiy@gmail.com

***Abstract*—Text coherence is an important factor that often gets overlooked by novice writers. Incoherence in academic writing directly affects both the reading experience and the comprehensibility of the articles. This paper introduces and describes a method for detecting incoherence in academic writing. The method utilized a fine-tuned BERT model in conjunction with a graph clustering algorithm. We benchmarked the method against baseline models on Discordant Sentence Detection using Time-travel dataset, and the results showed that the proposed method outperformed baseline models in terms of F1-score. Afterwards, the method was tested on corpora of Russian and English scientific articles in order to assess its proficiency in Narrative Incoherence Detection when applied to the paper's main research subject: academic writing. The paper's proposed method achieved a decent F1 of over 0.65 in Discordant Sentence Detection. For future work, our biggest goal is to further refine the method and be able to effectively deploy it on existing systems for reviewing academic corpora**

I. Introduction

In most, if not all cases, sentences in a text are not standalone units but rather interconnected entities that constitute a coherent paragraph. When sentences in the text are not logically connected - incoherence occurs. Distinguishing a coherent text from incoherent ones has been one of the key problems in discourse analysis. This very problem that the research team was tackling in this paper has been studied in different forms with different proposed tasks, including narrative cloze tasks [1][2]; discourse relations, sentence position, binary sentence ordering, discourse coherence, and sentence section prediction [3]; narrative incoherence detection [4].

Incoherence occurred during academic writing would create difficulties in conveying and spreading the authors' ideas. Such is the case for amateur writers such as bachelor students who are writing their graduation thesis, they often fail to produce sentences with strong interconnectivity. Which, to put it simply, means that their writings do not come together as a whole and the paragraphs feel more like a collection of sentences loosely put together. This notion allows the research team to theorize that the lack of coherence between sentences would manifest itself as a research article being "Poorly written" and "Difficult to follow" [5]. This lack of coherence could be the result of both intentional and unintentional actions. For example, should a student, when writing their research, verbate from an existing work, it is highly possible for the copied part to be incoherent when compared to the rest of the student's research. This would be considered as an intentional action, whereas any issue that arises from the writers' writing capability would be considered unintentional actions.

Unfortunately, most existing systems that provide common error verification for scientific papers [6] are quite lacking in terms of advanced quality checking features. In our previous work [7], we developed a model with the goal of enhancing existing systems by providing the ability to check the contextual relevance of keywords. As a follow-up, this work is aimed to provide the aforementioned error verification systems with discourse coherence checking ability. Thus, the result of this paper can serve as a valuable addition for improving existing systems, further assisting human reviewers in the verification process and stepping closer to automating the entire verification process.

This paper focuses on solving the Discourse Coherence task [3] and Narrative Incoherence Detection task [4]. For example, as illustrated in Fig.1, we would take a paragraph of text and determine if the paragraph is coherent or not. Additionally, we would isolate and locate the sentences that cause the incoherence. For that, we propose a method for detecting incoherent sentences in a paragraph using a fine-tuned BERT model [8] and a simple clustering algorithm. Moreover, the main research subject of this paper would be the corpora consisting of scientific articles written in English and Russian.

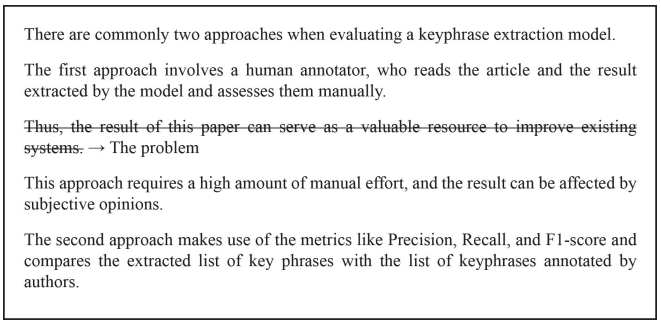

Fig. 1. Example of incoherent paragraph

In the next section, we summarize related research literature. Section III describes our method for detecting incoherent sentences and missing sentences. Section IV and V, respectively, present the setups of our experiments and the experiment results. Finally, Section VI presents the result and conclusion of the paper as well as suggestions on possible future research and practical application of this paper.

II. Related work

Over the years, a number of coherence modeling techniques have been proposed. Earlier works presented entity-based models for assessing text coherence [9]. Using this method, discourse entities are populated along with their grammatical roles in a grid, which is called an "entity grid". Later, [10] proposed a neural version of the entity grid model where they utilize a convolutional network to analyze the text and compute coherence score. The entity grid is converted into a feature vector convenient for feeding into a neural network, which assesses the discourse's coherence. The release of BERT [8] in 2018, and subsequently its application in the BERT-enhanced Relational Sentence Ordering [11], allowed for improved coherence modeling via enhancing the capturing of dependency relationship between sentences. All of the aforementioned models focus on assessing the discourse coherence of the text by focusing on solving the Sentence Ordering Task without pointing out the position of incoherent sentences that breaks text coherence, which are also known as discordant sentences according to [4]. Thus, [4] fixes the problem of the unidentified discordant/incoherent sentences by creating the Narrative Incoherence Detection task. They also propose two baseline models for solving the task (Token-level and Sentence-level models). Both of which utilize fine-tuned BERT models for solving the task. However, the token-level approach requires feeding the whole text (paragraph) at once to the model, which needs a tremendous amount of memory as the BERT attention mechanism scales quadratically to the sequence length. The sentence-level approach fixes the computation and memory cost problem by first using BERT to calculate sentence embeddings for all sentences in the text, then employing a second BERT-sized model to classify those sentences. This method fixes the runtime rise in memory usage; however, it still requires loading two BERT-sized models to the memory when applied in production. To solve this memory problem, we only used one BERT-sized model as well as feeding the model two sentences per input instead of the whole paragraph in order to avoid a rise in memory usage.

Previous works have studied discourse coherence in various domains: [4] used the TripAdvisor dataset, a collection of hotel reviews and TimeTravel dataset that contains stories on daily life; [3] utilized datasets compiled from Wikipedia and Ubuntu Internet Relay Chat. Despite the abundance of topics in the datasets that were utilized by the aforementioned studies, scientific articles were nowhere to be found in those datasets. Thus, this raises certain concern for the research team regarding the accuracy of the incoherence detection modules from previous works when applied to scientific corpora. On the other hand, previous works did not publicly publish their model and code, making it hard to reproduce their results. Therefore, our research team proposes an alternative method, aiming to produce competitive results and specifically inspect the performance when applied for detecting incoherence in the discourse of scientific articles. Furthermore, we publish our model and source code at [12] for other researchers who are interested in our work can easily adapt to solve their own problem.

III. Proposed model

Released in 2018, BERT was regarded as a state-of-the-art model due to its result when performing several natural language processing tasks. Based on BERT, a number of models have been developed to be faster and more accurate when doing these tasks. However, for our research, we decided to use the basic BERT model as our base due to two main reasons: the first is that BERT is highly accessible via the Huggingface library [13]; and the second reason is the fact that many transformers models based on BERT ignored the Next Sentence Prediction training objective, which was experimentally proven in BERT original paper [8] to hurt performance in multi-sentence tasks (tasks that require BERT to find the relationship between two or more input sentences). Due to this, our method for detecting incoherent sentences is built on top of the output of a fine-tuned BERT model used for the Next Sentence Prediction task. Therefore, to describe the method in detail, we split this section into two subsections: the first subsection presents the fine tuning process of the BERT model; the second subsection presents our algorithm for composing the desired result from the BERT model output.

A. Fine-tuning BERT model

To further explain the fine tuning process, it is necessary to discuss the details regarding how the model treats input data. First, a pair of sentences is fed into the BERT model. Afterwards, the high dimensional vector output of the BERT model is passed through a classification layer, which then outputs a confidence score representing the probability that the second sentence of the input pair directly follows the first sentence. In order to fine-tune the model, the following steps were taken:

1) Construct samples for fine-tuning: First, we needed to construct the sample set from the dataset. Since the BERT model takes input in this formula of: [CLS] + First sentence + [SEP] + Second sentence + [SEP] + Paddings, it was vital that each of our samples consisted of two sentences. Positive samples were formed by taking two sentences next to each other, while negative samples were created by applying one of the three following strategies randomly:

- Both the first and second sentence are taken randomly from two different paragraphs in the corpus. To minimize the probability of the two sentences being logically consistent, the two paragraphs are taken from different segments of the article.
- Two consecutive sentences are taken from a paragraph from the corpus (similar to the positive cases) but their order is reversed.
- One sentence is taken from the corpus and is used for both the first and second sentences (duplicate sentences).

The strategies for creating negative samples were used specifically in order to train the model's ability to distinguish correct-ordered sentences from random sentences, duplicate sentences and reverse-ordered sentences. The effect of the aforementioned strategies could be seen most clearly during the matrix creation step, which would be discussed later in the paper.

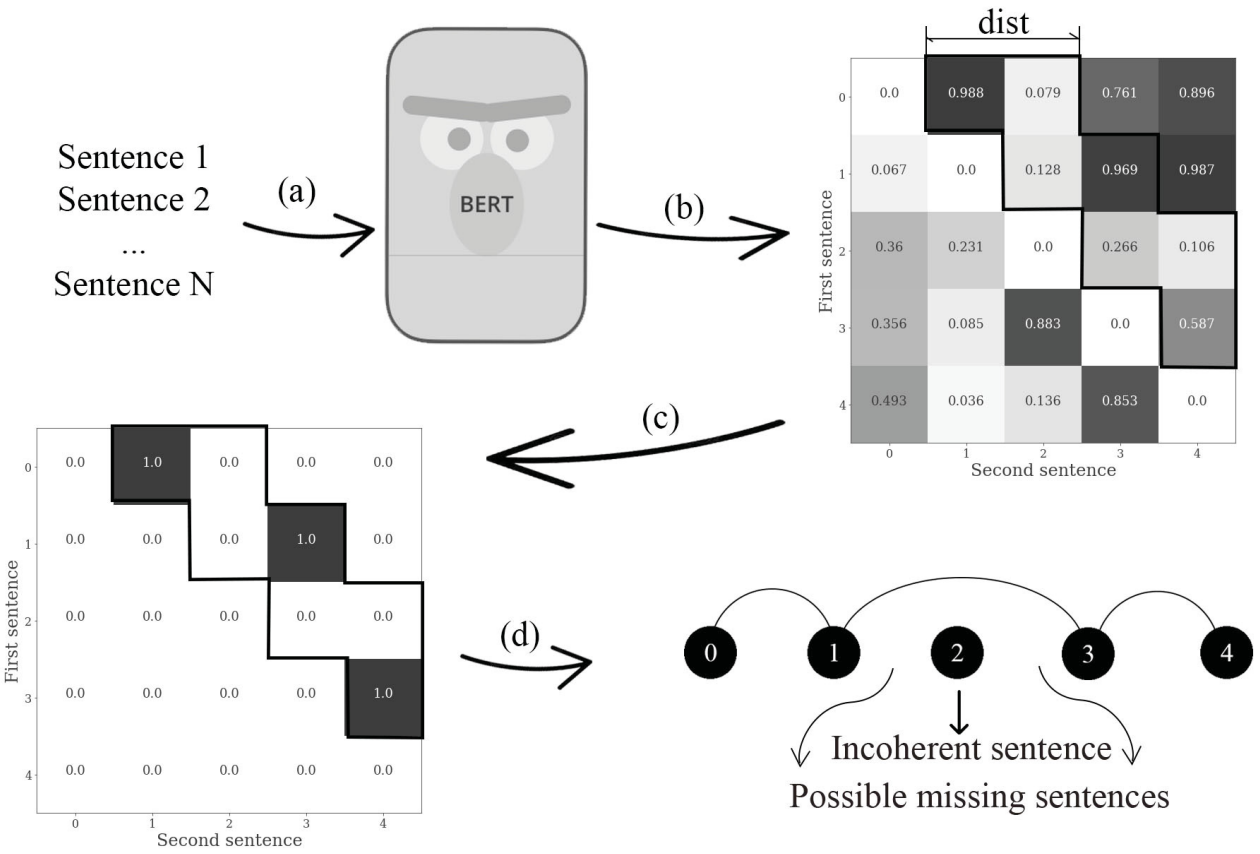

Fig. 2. Visualization of Incoherent Sentence and Missing Sentence detection process (a) The paragraph is fed through the BERT model as sentence pairs; (b) The result is then populated on a matrix; (c) A mask is applied on to the matrix to filter out irrelevant information; (d) A spanning forest is built and the conclusion is made

The number of positive and negative samples are kept at a balanced ratio. We assigned the golden scale label of '0' for positive samples and '1' for negative samples. These outputs will be used in the next step.

The inspected datasets, which would be described in greater detail in Section IV, were put through a process that filtered and classified the articles in terms of paragraph count. These articles were applied to the samples construction process, which is executed automatically using a Python script [14]. This script can be used to create samples for fine-tuning the model on any dataset of similar structure. This way, theoretically, we can adapt this model to different domain areas with relative ease.

2) Fine-tune the model: In order to fine-tune the BERT models, the research team followed the progress described in [8]. Firstly, the input sentences were tokenized, then the [SEP] tokens were added after each sentence and the [CLS] token was added at the beginning of the first sentence. Finally [PAD] tokens were inserted at the end of the second sentence in order to reach the desired batch length of 128 tokens. Although the sequence length limit for the BERT model is 512, for our experiments, we chose a batch length of 128 to speed up the training process This is highly sufficient based on the fact that the mean number of tokens for sentence pairs in our datasets did not exceed 60 tokens - barely half of the chosen value. The received vectors of integers were then fed to the BERT model. After feeding the samples into the model, the token embedding vector output that corresponded to the [CLS] token was then fed to a linear layer with a single output, representing the probability that the second sentence followed the first sentence. The cross-entropy loss was calculated by comparing the probability output to the golden scale label; and afterwards an optimizer was used to fine tune the model.

B. Algorithm for Narrative Incoherence Detection

The overall process for the algorithm is visualized in Fig.2. The process begins with the fine-tuned model calculating confidence scores on all sentence pairs in a paragraph, then arranges them on an N by N matrix, with N being paragraph length in sentence count. It is easy to observe that the diagonal line of the matrix is completely white, while the upper right half is darker in color compared to the bottom left half. This result is in-line with the research's intention during the creation of the samples.

Coherence is achieved when sentences are structurally and logically connected. In our work, for a paragraph to be classified as fully coherent, we use the following definition: each sentence in the paragraph must be related to at least one other sentence in the same paragraph. Due to the inter-connectivity between sentences of the same paragraph, the

research team hypothesized that the distance between two inspected sentences could only diminishingly affect the model. Specifically speaking, after the aforementioned distance has reached a certain threshold, which is called the "maximum distance", any further increase in distance would be negligible. Consequently, we applied the following mask function on the probability matrix to filter out the irrelevant information:

$$B(i,j) = \begin{cases} 1 & \text{if } A(i,j) > threshold \\ & \text{and } 0 < j - i <= dist \\ 0 & \text{otherwise} \end{cases}$$

where i is the first sentence index, j is the second sentence index, $threshold$ is the rounding limit and $dist$ is the maximum distance between inspected sentences.

The mask function only returns value for cells in the upper right half of the matrix that are, at the very most, dist away from the main diagonal line. The function returns '1' for the cells containing probability values greater than a threshold and '0' for the rest of the cells. For example, in Fig.2, with dist set to 2, the model only reads the cells inside the bold line while ignoring anything outside. In our experiments, the threshold value is empirically set to 0.5 while dist is further inspected through the experiments described in Section V.

The last step of the algorithm is to determine the position of incoherent sentences or possible missing connecting sentences. Treating the result matrix as an adjacency matrix, we created a graph with sentences as vertices. We then applied a simple algorithm to find spanning trees on the graph. In our experiments, we used the Kruskal algorithm [15] for this task. Initially, each sentence is in a separated tree. We looped through the adjacency matrix and for each edge (represented by a cell with value of 1), we merged the two trees which contain the two endpoints. After applying the algorithm, we received a spanning forest. In the case of a coherent paragraph, the spanning forest would become a single spanning tree. Otherwise, there would exist multiple tree clusters where a single vertex tree is an incoherent sentence. Between two different tree clusters, it is possible that there are missing connecting sentences.

IV. Experiment setup

This section will describe the setup of our experiment, including the dataset and base models used in the experiments as well as the metrics used during evaluation.

A. Dataset

1) Arxiv Dataset: Arxiv is a free distribution service and an open-access archive for scientific articles. In order to train and test our method on input articles written in the English language, scientific literature on Arxiv was fetched and compiled into datasets through the use of a script. Due to our faculty's field of research being Computer Science and Technology, the research team decided to fetch articles in the field of Computer and Information Science on Arxiv since the two fields were similar. However, the e-prints on Arxiv are not peer-reviewed which reduces their reliability. Therefore we had to only take articles with Journal-ref which were articles that had appeared in paper journal prints.

To avoid breaking the paragraphs when extracting from PDF, we first downloaded the Latex source of the articles, compiled it to HTML, and extracted only the text in paragraphs, leaving tables, figures, and section titles as well as reference lists. This simple yet effective method allowed for a dataset of relatively high quality.

For future researchers, we have also published the dataset along with building scripts at [14]

2) TimeTravel Dataset: TimeTravel is a dataset containing self-contained five-sentence stories focusing on commonsense and daily life. It was first introduced in [16]. In this paper, we used this dataset only to benchmark our method against baseline models introduced in [4].

3) Cyberleninka Dataset: For Russian, we used the dataset which we composed from our previous work [7]. The dataset contains scientific articles in the field of Computer and Information Science. The original purpose of the dataset was as inputs for keyword extraction tasks; however, it also contains the full text of the articles, split by paragraphs, which is useful for our current task.

For all datasets, we only retained paragraphs that are longer than two sentences. Except for the TimeTravel dataset, which has a fixed paragraph length of 5, the paragraph lengths in Arxiv and Cyberleninka dataset are varied. The paragraph length distribution after filtered out short paragraphs is populated in Table I. In addition, each dataset is split into a training set (90%) and a test set (10%). To improve consistency between runs, the training samples are created dynamically while the test samples are created only once during data preprocessing.

TABLE I. Paragraph length distribution for English and Russian dataset

Sentence per paragraph	Russian	English
3	47.02%	27.47%
4	25.31%	22.00%
5	12.88%	16.47%
6	6.81%	11.24%
7	3.82%	7.42%
8	1.92%	5.18%
9	0.99%	3.36%
10	0.54%	2.28%
11	0.29%	1.56%
12+	<1%	

Summary of the information on the datasets are populated in Table II

TABLE II. Dataset information summarization

	Language	Training	Test	Average paragraph per article	Average sentence per paragraph	Average word per sentence
Cyberleninka Dataset	Russian	1888	210	11.9	4.1	17.0
Arxiv Dataset	English	1423	158	28.3	5.3	22.0
TimeTravel	English	125229	7484	1.0	5.0	8.8

B. Evaluation Metrics

As our experiments are binary classifications, we will use Precision, Recall, F1, and Accuracy as metrics for evaluation following common practice. A higher score indicates better performance for all the metrics.

C. Pre-trained BERT models and fine-tuning

We used the BERT model implementation and pre-trained models from Huggingface [13] and DeepPavlov [17] for the experiments.

From Huggingface:

- BERT-base-uncased [8] and BERT-large-uncased [8] are base models trained for English corpus.
- BERT-base-multilingual [8] is the multi-language version of BERT-base

From DeepPavlov:

- RuBERT-base [18] is trained on Russian corpus.

Adam optimizer was used to train the model. To search for the optimal hyper-parameters, we performed a grid search for learning rate [2e-5, 5e-5] and batch size from [8, 16, 32] as recommended in the original paper. In the end, a learning rate of 2e-5 and batch size of 32 was chosen.

The experiments are conducted on PyTorch framework [19]. To reduce the effects of randomness in training, the final results are taken as the average results over 5 random initializations.

V. Experiments and results

This section features the results of our experiments. The experiments had three tasks: Next Sentence Prediction (NSP), Narrative Incoherent Detection (NID), Paragraph Classification (PaC). The NSP task was used in order to evaluate the performance of different models when doing basic fine-tuned tasks. The result of the NSP task is presented in Subsection V-A. The NID task was used for the purpose of assessing the performance of our model when locating incoherent sentences among the body of text. Subsection V-B and Subsection V-C, respectively, present the result of the experiment where we applied our model on the benchmark dataset (TimeTravel) and our datasets of scientific articles. Finally, the PaC task is an extension of the NID task with input leveraged to paragraph level. The description, details and result of the experiment regarding PaC tasks are presented in Subsection V-D.

A. Next Sentence Prediction

NSP is one of the two pre-training objectives of the BERT model. In this task, the model was fed two input sentences simultaneously, and it had to determine whether or not the second sentence followed the first sentence. Since the pre-trained models were trained on a dataset with contents of varying topics, we fine-tuned them using our chosen dataset of the relevant topic. The sample creation process is described in Section III-A. For this experiment, we used Accuracy as the evaluation metric.

From the result populated in Table III, we can see that on English datasets, BERT-large-uncased performed best while on Russian dataset, RuBERT-base-uncased performed best. We will investigate how these two models perform on NID and PaC tasks in succeeding experiments.

TABLE III. Next Sentence Prediction experiment result

Name	Language	Dataset		
		Arxiv	Cyberleninka	TimeTravel
BERT-eng	English	0.830		0.916
BERT-eng-large	English	0.859		0.919
BERT-multilingual	Multilingual	0.843	0.826	
RuBERT	Russian		0.848	

B. Narrative Incoherence Detection benchmark

The benchmark consists of two scenarios: Missing Sentence Detection (MSD) considers the cases where some semantic gaps are caused by missing connecting sentences and Discordant Sentence Detection (DSD) considers the cases where sentences in a paragraph are incoherent to the surrounding context.

For benchmarking, we conducted an experiment similar to the experiment described in [4], then compared our results with their published results. The TimeTravel dataset, which consists of five-sentence stories, was used as input for both of the benchmark scenarios. For the MSD scenario, one random sentence in each of the stories was removed. Whereas for the DSD scenario, we first had to choose a confounding sentence using the procedure described in [4] and then replace one random sentence in each of the stories with that confounding sentence. The procedure employed to choose a confounding sentence could be described in two steps: First, the top 100 most similar sentences from the entire dataset were selected using the fast BM25 retrieval [20]. Then, the second step is to choose the first sentence in the returned list that has $sim(a, b) < \tau$, where a is the original sentence; and b is the confounding sentence; similarity sim is measured by BERTScore [21] and τ is empirically chosen to be 0.7.

As we can see in Table IV and Table V, with appropriately chosen dist value, our model outperformed the baseline models [4] and achieved a higher F1 score in DSD scenario. However, in the MSD scenario, we were not able to surpass the baseline. Despite that, we achieved a slightly higher Precision score than the baseline model, which can be interpreted that the errors that we located are more accurate, but we cannot locate as many errors as the baseline.

TABLE IV. Benchmark result (Missing Sentence Detection)

		Precision	Recall	F1	Accuracy
Our model	dist=1	0.606	0.326	0.424	0.704
	dist=2	0.639	0.210	0.317	0.697
	dist=3	0.627	0.167	0.264	0.689
	dist=4	0.633	0.167	0.265	0.690
	dist=5	0.634	0.170	0.269	0.690
Baseline	Token-level	0.616	0.552	0.582	0.736
	Sentence-level	0.594	0.431	0.500	0.712

A lower score on the MSD task is expected since the MSD task requires a deep understanding of relation over the paragraph. The baseline models achieved this knowledge by utilizing the transformer model on the paragraph level. In contrast, our method is strictly based on next sentence classification - in other words, it relies on the probability of two

TABLE V. BENCHMARK RESULT (DISCORDANT SENTENCE DETECTION)

		Precision	Recall	F1	Accuracy
Our model	dist=1	0.593	0.903	0.716	0.799
	dist=2	0.745	0.849	0.793	0.875
	dist=3	0.815	0.810	0.812	0.896
	dist=4	0.830	0.803	0.816	0.899
	dist=5	0.828	0.802	0.815	0.898
Baseline	Token-level	0.632	0.624	0.628	0.852
	Sentence-level	0.611	0.479	0.537	0.835

sentences being next to each other. However, strictly adhering to next sentence classification allowed us to achieve better results compared to the baseline models in the DSD task.

C. Narrative Incoherence Detection in scientific articles

A similar experiment to the one described in the previous subsection was performed using our two chosen datasets of scientific articles as inputs. The result of the experiment, populated in Table VI and Table VII, is similar to the previous subsection's benchmark result.

TABLE VI. MISSING SENTENCE DETECTION IN SCIENTIFIC ARTICLES

	Distance	Precision	Recall	F1	Accuracy
English	1	0.409	0.191	0.260	0.662
	2	0.446	0.088	0.147	0.682
	3	0.498	0.067	0.119	0.688
	4	0.518	0.062	0.111	0.690
	5	0.525	0.057	0.103	0.690
Russian	1	0.480	0.139	0.216	0.523
	2	0.518	0.081	0.140	0.531
	3	0.540	0.064	0.114	0.533
	4	0.558	0.061	0.109	0.534
	5	0.560	0.061	0.110	0.534

TABLE VII. DISCORDANT SENTENCE DETECTION IN SCIENTIFIC ARTICLES

	Distance	Precision	Recall	F1	Accuracy
English	1	0.566	0.776	0.655	0.797
	2	0.739	0.692	0.715	0.863
	3	0.798	0.653	0.719	0.873
	4	0.807	0.638	0.713	0.873
	5	0.817	0.619	0.704	0.871
Russian	1	0.564	0.896	0.692	0.768
	2	0.733	0.870	0.795	0.869
	3	0.776	0.839	0.806	0.882
	4	0.779	0.846	0.811	0.885
	5	0.787	0.830	0.808	0.885

We observe the followings: For DSD:

- The method produced a low F1 score with dist set to 1.
- For other values of dist, the metrics are similar, which backed our hypothesis that there will not be any observable increase in terms of impact after the distance between chosen sentences reaches a certain threshold during the DSD task.

For MSD:

- The highest F1 score is achieved when the distance parameter is set to 1.

In succeeding experiments on the Paragraph Classification task, according to this result, we set distance values for MSD and DSD, respectively, to 1 and 2.

D. Paragraph classification

In this experiment, we wanted to see how our method works on the paragraph level. PaC is a binary classification task, formulated similar to the Discourse Coherence task from [3]. The input for PaC is a paragraph of N sentences and the task is to determine if the paragraph is coherent or incoherent. The original proposed task only looked at the case when Discordant Sentence (DS) occurs. In our work, we examined two cases: the first one is when only the Missing Sentence (MS) scenario occurs; the second one is when only the DS scenario occurs. To create test instances for the task, we used the following strategy:

- The negative (as in no errors are found) samples are created from a full paragraph in the corpus.
- Positive samples in Missing Sentence cases are created by removing one sentence from the paragraph.
- Positive samples in Discordant Sentence cases are created by replacing one sentence from the paragraph with a confounding sentence.

We further leverage the PaC task by adding an Overall test. In this test, the model must determine if the paragraph falls into one of the four classes: No Error, Only MS, Only DS, Both Errors. In this test, we consider the No Error class as negative cases and the rest as positive cases. The samples used for No Error, MS, and DS classes are created the same way as described above, while for Both Errors class, we replace one sentence from the paragraph and, at the same time, remove another non-adjacent sentence. In both tasks, the number of instances is balanced between classes.

The result of the experiment is shown in Table VIII.

TABLE VIII. PARAGRAPH CLASSIFICATION RESULT

	Language	Precision	Recall	F1	Accuracy
Only Missing Sentence	English	0.569	0.331	0.419	0.540
	Russian	0.537	0.321	0.402	0.522
Only Incoherent Sentence	English	0.831	0.725	0.775	0.789
	Russian	0.844	0.828	0.836	0.837
Overall	English	0.892	0.767	0.825	0.747
	Russian	0.860	0.730	0.790	0.715

From the confusion matrix shown in Fig.3, we were able to observe that although our method can classify paragraphs containing errors and not containing errors with an accuracy of higher than 0.7 and an F1 score of higher than 0.8, it is not good at distinguishing error types:

- MS is often mistaken for DS or ignored and classified as No Error.
- DS is often classified as Both, which means DS is correctly detected; however, Missing Sentence is at the same time incorrectly detected.

The Overall test score can be interpreted as the system's ability to find paragraphs containing errors. With an F1 score of over 0.8, it is possible to use the method to assist human reviewers in evaluating scientific writing. However, as the method lacks the capability to distinguish between different types of errors, thus it is not recommended to use the method for automatic scoring systems.

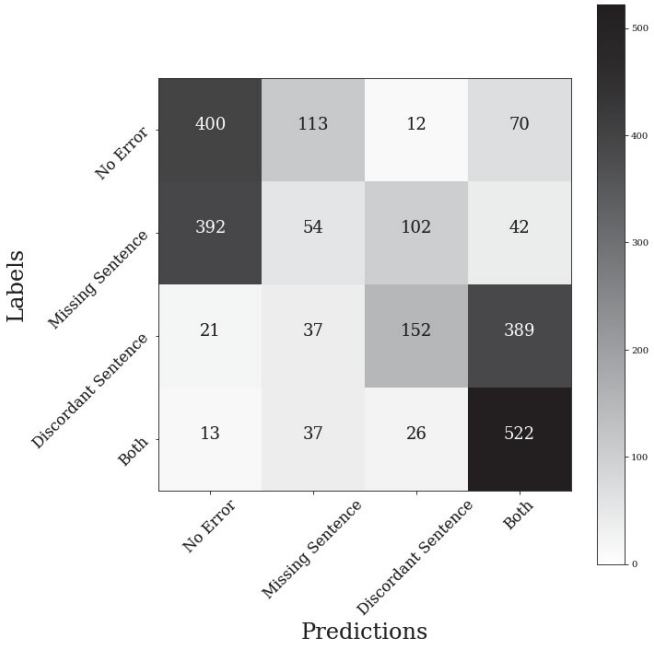

Fig. 3. Confusion matrix of the Paragraph Classification experiment

VI. Conclusions

In this work, we introduced, described and evaluated our model for incoherent sentence detection in scientific articles written in Russian and English. Our approach outperformed the chosen baseline on the Discordant Sentence Detection task. Although it achieved a higher Precision metric on the Missing Sentence Detection task, the F1 metric is lower than the baseline. We also tested our method on English and Russian corpus of scientific articles and achieved positive results. Our current objective would be deploying this model on existing systems for evaluating students' scientific writing [6] as well as enhancing the incoherence detection feature of this model. In the future, it would be of great interest to develop a better method for the Missing Sentence Detection task.

Acknowledgment

We thank Dat Khoa Nguyen for his support in the process of writing this article.

References

[1] N. Chambers and D. Jurafsky, "Unsupervised Learning of Narrative Event Chains", *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, Jan.2008, pp.789-797.

[2] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli and J. Allen, "A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories", *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jan.2016, pp.839-849.

[3] M. Chen, Z. Chu and K. Gimpel, "Evaluation Benchmarks and Learning Criteria for Discourse-Aware Sentence Representations", *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Jan.2019, pp.649-662.

[4] D. Cai, Y. Zhang, Y. Huang, W. Lam and B. Dolan, "Narrative Incoherence Detection", *arXiv:2012.11157 [cs.CL]*, Dec.2020.

[5] D. Pierson, "The Top 10 Reasons Why Manuscripts Are Not Accepted for Publication", *Respiratory care*, Oct.2004.

[6] E.I. Blees and M.M. Zaslavskiy, "Criteria for text conformity to scientific style", *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, vol.19, Apr.2019, pp.299-305.

[7] Q. H. Nguyen and M. Zaslavskiy, "Keyphrase Extraction in Russian and English Scientific Articles Using Sentence Embeddings", *Proceedings of the 28th Conference of Open Innovations Association*, Jan.2021, pp.334-340.

[8] J. Devlin, M. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *arXiv:1810.04805 [cs.CL]*, Oct.2018.

[9] R. Barzilay and M. Lapata, "Modeling Local Coherence: An Entity-Based Approach", *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, vol.34, Jan.2005.

[10] S. Joty, M. Mohiuddin and D. Nguyen, "Coherence Modeling of Asynchronous Conversations: A Neural Entity Grid Approach", *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol.1, Jul.2018, pp.558-568.

[11] B. Cui, Y. Li and Z. Zhang, "BERT-enhanced Relational Sentence Ordering Network", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Jan.2020, pp.6310-6320.

[12] Q.H Nguyen, ISDetection, Web: - https://github.com/levi218/ISDetection.

[13] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T.L. Scao, S. Gugger, M. Drame, Q. Lhoest and A.M. Rush, "Transformers: State-of-the-Art Natural Language Processing", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Oct.2020, pp.38-45.

[14] Q.H Nguyen, ArxivDataset, Web: - https://github.com/levi218/ArxivDataset.

[15] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem", *Proceedings of the American Mathematical society*, vol.7, 1956, pp.48-50.

[16] L. Qin, A. Bosselut, A. Holtzman, C. Bhagavatula, E. Clark and C. Yejin, "Counterfactual Story Reasoning and Generation", *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Jan.2019, pp.5046-5056.

[17] M. Burtsev, A. Seliverstov, R. Airapetyan, M. Arkhipov, D. Baymurzina, N. Bushkov, O. Gureenkova, T. Khakhulin, Y. Kuratov, D. Kuznetsov, A. Litinsky, V. Logacheva, A. Lymar, V. Malykh, M. Petrov, V. Polulyakh, L. Pugachev, A. Sorokin, M. Vikhreva and M. Zaynutdinov, "DeepPavlov: Open-Source Library for Dialogue Systems", *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Jul.2018.

[18] Y. Kuratov and M. Arkhipov, "Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language.", *arXiv:1905.07213[cs.CL]*, May 2019.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library", *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp.8024-8035.

[20] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond", *Foundations and Trends in Information Retrieval*, vol.3, Jan.2009, pp.333-389.

[21] T. Zhang, V. Kishore, F. Wu, K. Weinberger and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT", *arXiv:1904.09675 [cs.CL]*, Apr.2019.