DIPS at CheckThat! 2021: Verified Claim Retrieval

Simona Mihaylova[1], Iva Borisova[1], Dzhovani Chemishanov[1], Preslav Hadzhitsanev[1], Momchil Hardalov[1] and Preslav Nakov[2]

[1] *Faculty of Mathematics and Informatics,*
Sofia University "St. Kliment Ohridski",
Sofia, Bulgaria

[2] *Qatar Computing Research Institute, HBKU,*
Doha, Qatar

Abstract

This paper outlines the approach of team DIPS towards solving the CheckThat! 2021 Lab Task 2 – a semantic textual similarity problem for retrieving previously fact-checked claims. The task is divided into two subtasks, where the goal is to rank a set of already fact-checked claims based on their relevance to an input claim. The main difference between the two is the data sources, i.e., Task 2A's claims are tweets, while Task 2B – debates and speeches. For solving the task, we combine variety of algorithms – BM25, S-BERT, a custom classifier, and RankSVM into a claim retrieval system. Moreover, we show that data preprocessing is critical for such tasks and can lead to significant improvements in MRR and MAP. We have participated in the English edition of both subtasks and our system was ranked third in Task 2A, and first in Task 2B.

Keywords

Check-Worthiness Estimation, Fact-Checking, Veracity, Verified Claims Retrieval, Detecting Previously Fact-Checked Claims, Social Media Verification, Computational Journalism, COVID-19

1. Introduction

Claims can have consequences, especially now in the middle of the COVID pandemic fake news, dubious content, mis- and disinformation are becoming more and more influential [1]. There are a plethora of socially significant topics, that are objects of massive falsification that have already affected our day-to-day lives. Such topics include the virus origin, cures, vaccines, the effectiveness of the applied measures among many others. An indicative example is the fact that accidental poisonings from bleach and disinfectants have unprecedentedly risen after a single claim from the US political scene.[1] Such frivolous claims may seem harmless at first, but being said by public figures, they automatically gain popularity and thereby, influence. Moreover, it is not even necessary for the statements to be made by public figures as social media offer a platform for anyone to share their opinions and views.

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

smartinovm@uni-sofia.bg (S. Mihaylova); ivadb@uni-sofia.bg (I. Borisova); dchemishan@uni-sofia.bg (D. Chemishanov); hadzhicane@uni-sofia.bg (P. Hadzhitsanev); hardalov@fmi.uni-sofia.bg (M. Hardalov); pnakov@hbku.edu.qa (P. Nakov)

[1] https://www.forbes.com/sites/robertglatter/2020/04/25/calls-to-poison-centers-spike–after-the-presidents-comments-about-using-disinfectants-to-treat-coronavirus

The rapid rate at which the information is spread nowadays calls for the development of better systems for detecting such potentially harmful content. These systems eventually either can be a tool for professional fact-checkers, or in the long term they can even operate in a fully automated manner.

The first few minutes of spreading a claim are key for its virality [2], and therefore it is important to disprove any false claims as fast as possible. The volume of statements during a live stream or in the news feed of social media is such that it is not possible to do it manually and machine help is needed to let fact-checkers concentrate on the claims that have not been seen and fact-checked by anyone so far.

Our work focuses on the CheckThat! 2021's task for claim retrieval. In particular, its aim is to alleviate the fake news detection process by providing a mechanism for fast fact-checking of a given claim. The task is to rank a set of already fact-checked claims by their relevance to a given input text, which contains a claim.

The task [3] consists of two subtasks, offered in English and Arabic, and we participated in 2A and 2B only for English. The tasks are defined as follows:

- **Task 2A:** Given a tweet, detect whether the claim the tweet makes was previously fact-checked with respect to a collection of fact-checked claims. This is a ranking task, where the systems will be asked to produce a list of top-N candidates.
- **Task 2B:** Given a claim in a political debate or a speech, detect whether the claim has been previously fact-checked with respect to a collection of previously fact-checked claims. This is a ranking task.

Our experimental setup bears close resemblance to the one proposed by Shaar et al. [4]. We tackled the problem using sentence BERT (S-BERT) [5], combined with other techniques, such as RankSVM [4, 6] to handle the re-ranking and a classifier neural network accepting the S-BERT scores as input. A standard practice for such tasks that we have adopted is to perform data preprocessing [7].

Our pipeline consists of the following steps:

1. Data preprocessing: extract Twitter handles as names and split the hashtags;
2. Compute BM25 scores for the given input claim and the fact-checked claims;
3. Compute the S-BERT embeddings in order to assign scores by calculating the cosine similarities between the input claims and the fact-checked claim candidates;
4. Pass these scores as an input to a classifier;
5. Pass the BM25 and S-BERT scores to RankSVM and obtain the final results.

The official results for our submission to the competition are mean reciprocal rank (MRR) of 0.795 for Task 2A and of 0.336 for Task 2B, and include items 1–3 from the list above. After the official competition deadline, we improved our results further to 0.909 for Task 2A and to 0.559 for Task 2B by experimenting with steps 4 and 5 above.

The rest of the paper is organized as follows: Section 2 discusses related work, Section 3 presents and explores the dataset, Section 4 introduces the methods we use, Section 5 describes the experiments and analyzes the results, and Section 6 concludes and discusses possible directions for future work.

2. Related Work

In recent years, there has been a growing interest in detecting mis- and disinformation [8, 9, 10, 11]. Fact-checking is one of the key components in the process. It was also part of the previous CheckThat! lab editions [12]. The majority of prior studies, including the CheckThat! 2020 winners' team for Task 2, which is similar to Task 2A of the 2021 edition - Buster.AI [7], used BERT architectures [13]. The Buster.AI team cleaned the tweets from non-readable input and used a pretrained and fine-tuned version of RoBERTa to build their system. They used a binary classification approach on the original tweet–claim pairs from the dataset and also on false pair examples, which they generated using various sampling strategies. Furthermore, they performed data augmentation with SciFact [14], FEVER [15], and Liar [16] datasets and tried Named Entity Recognition and Back and Forth Translation as additional enhancements. Other methods used by the teams in the competition include relying on cosine similarity for computing the scores corresponding to each claim, using Terrier [17] and Elasticsearch,[2] and performing data cleanup by removing URLs, hashtags, usernames, and emojis from the tweets.

Our experimental setup builds on the one proposed by Shaar et al. [4], which consists of BM25 implementation for finding the highest scores, which are then fed to RankSVM along with the S-BERT similarity scores. The experiments that have been conducted as part of their work showed significant improvements over state-of-the-art retrieval and textual similarity approaches. Furthermore, they created specialized datasets by retrieving data from the websites of PolitiFact and Snopes. Our pipeline differs from theirs mainly in the data handling: we use specific data preprocessing, which is described in Section 4.1, and we also use the *Date* field present in this year's competition dataset. Moreover, we significantly reduce the training time for the RankSVM by using a linear kernel instead of RBF without much impact on the results.

3. Dataset and Features

Below, we discuss the structure and the size of the datasets, and the features we use.

3.1. Task 2A: Detect Previously Fact-Checked Claims in Tweets

For Task 2A, the organizers collected 13,825 previously-checked claims related to Twitter posts. The claims are obtained from Snopes – a fact-checking site that focuses on "discerning what is true and what is total nonsense"[3].

Each fact-checked claim in the dataset is described by the following fields:

- *Title*: title of the article from which the fact-checked claim is extracted;
- *Subtitle*: subtitle of the article;
- *Author*: author of the article;
- *Date*: date on which the claim was fact-checked;
- *Vclaim_ID*: unique identifier of the fact-checked claim;
- *Vclaim*: text of the fact-checked claim.

[2] http://www.elastic.co/elasticsearch/

[3] http://www.snopes.com/

A total of 1,401 Twitter posts were provided for input claims. The fields for each input claim are as follows:

- *Iclaim_ID*: unique identifier of the input claim;
- *Iclaim*: text of the input claim.

Table 1
Number of *(input claim, checked claim)* pairs in the training, the development, and the testing sets for both tasks.

Task	Train	Dev	Test
Task 2A: Detect Previously Fact-Checked Claims in Tweets	999	200	202
Task 2B: Detect Previously Fact-Checked Claims in Political Debates/Speeches	562	140	103

For training, validation, and testing of the models, the organizers provided gold files containing the relevant *(input claim, checked claim)* pairs. Table 1 shows the number of examples in each of the sets.

3.2. Task 2B: Detecting Previously Fact-Checked Claims in Political Debates/Speeches

The verified statements for Task 2B are selected from articles confirming or refuting claims made during political debates or speeches. Their number is 19,250. The data is retrieved from PolitiFact[4] - a fact-checking website that rates the factuality of claims made by American politicians and elected officials.

The fact-checked claims contain the following columns:

- *URL*: the link to the article providing justification for the fact-checked claim;
- *Subtitle*: the subtitle of the article;
- *Speaker*: the original speaker or the source for the fact-checked claim;
- *Vclaim*: the text of the fact-checked claim;
- *Truth_Label*: the truth verdict given by the journalists;
- *Date*: the date on which the claim was fact-checked;
- *Title*: the title of the article providing justification for the fact-checked claim;
- *Vclaim_ID*: unique identifier of the fact-checked claim;
- *Text:* the text of the article providing justification for the truth label of the fact-checked claim.

The input claims are extracted from transcripts taken from political debates or speeches; their number is 669. Each input claim is described by the following fields:

- *Iclaim_ID*: unique identifier of the input claim;
- *Iclaim*: text of the input claim.

[4] http://www.politifact.com/

Table 2
Statistics about the dataset for Task 2A: Detecting Previously Fact-Checked Claims in Tweets.

Field	# of words				# of sentences			
	Min	Median	Mean	Max	Min	Median	Mean	Max
Iclaim	1	38	42.5	139	1	2	2.5	9
Title	1	11	10.3	29	1	1	1	3
Subtitle	0	20	19.8	51	0	1	1	4
Vclaim	1	18	19.1	122	1	1	1	6

Table 3
Statistics about the dataset for Task 2B: Detecting Previously Fact-Checked Claims in Political Debates/Speeches.

Field	# of words				# of sentences			
	Min	Median	Mean	Max	Min	Median	Mean	Max
Iclaim	2	20	23.2	107	1	1	1	2
Title	2	13	13.3	36	1	1	1.1	4
Vclaim	2	21	22.5	94	1	1	1.2	11
Text	117	999	1,029.4	11,008	3	41	42.3	273

3.3. Analysis

Due to the fact that the BM25 scores are calculated based on an exact match between the words in the query claim and the words in the document (the fact-checked claim), we analyzed the number of words in each field that contains context. Furthermore, we gathered statistics about the number of sentences in a claim as the S-BERT scores are obtained by comparing the entire query to each sentence from the fact-checked claims semantically.

The fields *Title*, *Subtitle*, *Vclaim* and *Iclaim* bring the semantics for Task 2A. Table 2 shows summarized statistics. We see that the mean number of words in *Iclaim* is approximately twice as high as in *Vclaim*. Thus, we can assume that if we add the words from *Title* and *Subtitle* to *Vclaim*, we might get a higher matching score with the query *Iclaim*. Similarly, comparing the results for the number of sentences, we can conclude that appending the *Title* and the *Subtitle* to the *Vclaim* would increase the level of details. However, there are fact-checked claims for which the *Subtitle* is empty, and thus it would not be meaningful to compare the claims by this field. From the data for Task 2A, we can conclude that if most of the words in the input and the fact-checked claim match exactly, we expect to achieve high results even with BM25.

For Task 2B, the important fields are *Iclaim*, *Title*, *Vclaim*, and *Text*. Table 3 shows that if we compare *Iclaim* by number of words or sentences only with *Vclaim*, or with *Vclaim* and *Title*, the values are relatively close. There would be a problem if *Title* and *Vclaim* do not contain enough information to confirm or to disprove *Iclaim*. Then, it would be reasonable to also use the *Text* field. However, this field is very long and has 41 sentences on average: about 40 times more than the sentences in *Iclaim*. Because S-BERT gives a separate score based on each sentence of the fact-checked statement, in order to achieve a good result, it is necessary to have a sentence in the *Text* field that is semantically close enough to the entire input claim.

Observing the data, we noticed that in Task 2A, at the end of each *Iclaim* field, the date of publication of the tweet is written. It is even in the same format as the date in the *Date* field for the fact-checked statements. Therefore, we consider that the *Date* field can be used to add more detail to the fact-checked claims. In Task 2B, we have *Date* field for the previously-checked claims and also observe that the field *Iclaim_ID* contains the date of holding the debate or the speech from which the notes are taken. Therefore, the date can be extracted from the IDs and added to the content of the respective input claim.

3.4. Back-Translation

We tried to exploit the fact that the task includes two languages in order to extend our training data. We used machine translation and attempted to triple the size of the original data by creating two additional datasets for the English language track, where we focused our efforts. Our goal was to have an English translation of the Arabic data and back-translation of the English data through Arabic and back. For this purpose, we used machine translation models pretrained on the OPUS corpus [18] and made available via the HuggingFace's Transformers library [19]. Though the translation was relatively straightforward, it took a long time to complete, and we did not manage to implement it fully eventually. Overall, the results from our experiment that include the back-translation technique seemed inconsistent, as they had large discrepancies depending on the test data. Two separate runs from our experiments with back-translation provided different results - the first one showed a slight improvement, while the second one slightly worsened the performance. At the end, the limited time frame for the competition was insufficient for us to investigate thoroughly what was happening, and thus we decided not to include back-translation in our submission.

4. Method

The approaches we used for our task are based on previous research in the field and on methods proven to be effective [4, 7, 12]. We gathered the knowledge regarding the state-of-the-art methods for solving the semantic sentence similarity task and incorporated them in order to improve the overall performance. However, due to the fact that there is no single method proven to perform significantly better than the rest, we conducted experiments with each approach separately, evaluated it, and finally, ensembled them into a unified pipeline. Figure 1 shows the workflow of our experiments. Below, we describe the elements of this pipeline.

4.1. Data Preprocessing

Before feeding the data into the models, we performed data preprocessing. In Task 2A, due to Twitter posts containing a lot of tags and usernames, we divided each PascalCase or camelCase sequence into individual words in order to add more context. PascalCase is a way to combine words by capitalizing all of them and removing the space and the punctuation marks, e.g., *DataPreprocessing*. camelCase combines a sequence of words by capitalizing all words except for the first one and removing the space and the punctuation marks between them, e.g., *dataPreprocessing*. For example, the tag `#NewYear2019` will be split as *New Year 2019*.

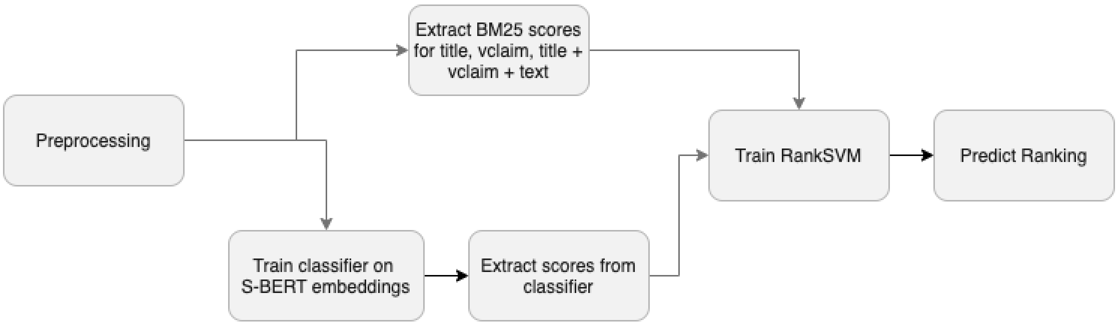
Preprocessing
Extract BM25 scores for title, vclaim, title + vclaim + text
Train classifier on S-BERT embeddings
Extract scores from classifier
Train RankSVM
Predict Ranking

Figure 1: The workflow of our experiments.

Some of the tweets contain a reference to another post or picture, which we consider unnecessary. Thus, we removed any sequence of characters that starts with `http` or `pic.twitter.com`. We also removed the emojis. Furthermore, we used the *Date* field from the dataset in order to add more detail. We experimented with just concatenating the *Date* to the end of the *Vclaim* field and also with inserting it as part of the last sentence.

For Task 2B, we also split PascalCase and camelCase words into individual strings. Due to the more formal style of political debates and speeches, there were no URLs or emojis. Furthermore, we noticed that in the *Text* field of each fact-checked claim file, the newline characters could reach about 15-20 consecutive occurrences, and thus we removed them. Similarly to Task 2A, we experimented with the *Date* field. First, we extracted the date from the fields *Iclaim_ID* and *Date*, and we converted both dates to the same format. Then, we used the strategies described for Task 2A to add the date to the respective input and fact-checked claims.

4.2. BM25 baseline

The organizers of the competition provided a simple BM25 baseline using Elasticsearch, which we used for our experiments. We improved over the baseline by using combinations of different dataset fields and applying data preprocessing.

For Task 2A, we conducted our BM25 experiment by combining the following fields:

- *Vclaim*
- *Vclaim* + *Title*
- *Vclaim* + *Title* + *Subtitle* + *Date*

For Task 2B, we used the following combinations:

- *Vclaim*
- *Text*
- *Title* + *Vclaim* + *Text*
- *Title* + *Vclaim* + *Date* + *Text*

Merging these fields, we aim to generate more detailed representation of the fact-checked claims. In Section 5, we elaborate on how featuring different fields impacts the results.

4.3. S-BERT

Previous research [4, 7] has demonstrated that neural networks are efficient for solving semantic textual similarity tasks. Our experimental design combines the sentence BERT (S-BERT) (`paraphrase-distilroberta-base-v1` [5]) model with a custom neural network. We constructed more detailed fact-checked claims by concatenating the fields *Title*, *Subtitle*, *Vclaim* and *Date* for Task 2A, and respectively the fields *Title* and *Text* for Task 2B. After performing simple preprocessing depending on the subtask, our system calculates the S-BERT embeddings separately for each text by splitting it into sentences and then comparing them using cosine similarity. The sorted list of these scores serves as an input to the neural network. We use the network architecture proposed in [4], which consists of an input layer, accepting as an input the S-BERT scores for the top-4 sentences, with 20 units and ReLU activation, a hidden layer with 10 units, ReLU, and an output layer with a sigmoid function. We used the Adam optimizer to optimize a binary cross-entropy loss. We hypothesise that the higher the number of sentences is, the noisier the data will be. We acknowledge that our experiments are insufficient to determine the exact impact, and we leave that for future work.

4.4. RankSVM

The BM25 baseline gives scores for exact matching for a given fact-checked claim and the input claim, while the S-BERT model compares them in a more semantic manner. To get the best of both techniques, we decided to use RankSVM[6] [4] with a linear kernel to re-rank the combined results. For both subtasks, we re-ranked the results yielded by the model that gives the highest mean reciprocal rank.

We performed the training of the model over the results obtained for the training dataset. For each pair *(input claim, checked claim)* in the top-N list from the selected model, we collected the corresponding scores from the *S-BERT* and the BM25 experiments with the respective reciprocal ranks. Then, we created a file containing these features in the proper format and we trained a RankSVM reranking model with a linear kernel to obtain a ranked list of the candidates. For each top-N *(input claim, checked claim)* pair, which are related according to a given gold file, we set the target value of RankSVM to be equal to 1, and for all the others we set it to 2. Then we re-ranked the test dataset using the trained RankSVM. The experiment is performed with different values for N and the search for an optimal N value is stopped when the results start to degrade.

For Task 2B, we also decided to include the *Date* field as a feature in the RankSVM reranker. For this purpose, we extracted the date from the field *Iclaim_ID* for each input claim and we turned it into seconds. After that, we converted into seconds the *Date* field for all fact-checked claims. Finally, for each pair *(input claim, checked claim)*, given to the RankSVM, we added a feature, composed of the logarithmic difference between the respective dates, measured in seconds.

[5] http://huggingface.co/sentence-transformers/paraphrase-distilroberta-base-v1

[6] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

5. Experiments and Evaluation

We describe our models and algorithms, the dataset fields we used, and the results achieved.

5.1. Evaluation Measures

Given that we have a ranking task, we use standard measures for ranking: Mean Reciprocal Rank (MRR) [20], Mean Average Precision (MAP), and Precision@k, ad MAP@k. Mean Reciprocal Rank is appropriate for cases when one relevant document is sufficient for the validation, e.g., the fact-checker may need only one certain proof to confirm or to reject a given claim. For scenarios where a more detailed overview is required, the Mean Average Precision measure and Precision@k are more suitable.

5.2. Model Details

Our model is implemented using PyTorch and Keras. We train the classifier for 15 epochs using a batch size of 2,048. For gradient accumulation, we chose Adam with a learning rate of 1e-3. For the Elasticsearch BM25 experiment, we used a query size of 10,000. When training the RankSVM with a linear kernel, we empirically established the appropriate value for the parameter C, which is the trade-off between training error and margin size. Usually, the optimal value was an integer between 3 and 10.

5.3. Results Task 2A (Tweets)

5.3.1. BM25

We performed experiments matching the *Iclaim* against the fields *Vclaim*, *Vclaim + Title*, and *Vclaim + Title + Subtitle + Date*. The performance of the model is tested on preprocessed and non-preprocessed data. Table 4 shows that the lowest performance with respect to MRR is achieved when combining *Vclaim + Title + Subtitle + Date* for non-preprocessed data: in this case, MRR is 0.744. This low score is most likely due to the fact that merging all fields adds a lot of redundant information. Using only the *Vclaim* yields a slightly better result for MRR: 0.752. We observed that adding *Title* to *Vclaim* increases the MRR by 0.09 because *Title* may have more words matching the given input claim than only the text of *Vclaim*.

The results in Table 4 show that preprocessing helps. Comparing to the experiment with the highest MRR for non-preprocessed data, we observe that after data preprocessing, MRR increases by 0.02—0.07 points absolute, with the best score achieved when combining all four fields. Adding *Subtitle* and *Date* to *Vclaim + Title* improves MRR by 0.03 points absolute, compared to using non-preprocessed data, where adding the two fields reduces MRR by 0.15 points absolute. We conclude that the sizable improvement when applying preprocessing is due to the fact that splitting the tags and the usernames adds more context, and removing URLs and emojis reduces the unnecessary information. Moreover, this proves that it is more efficient to insert the *Date* as part of the last sentence of the *Vclaim*, than to treat it as a separate sentence as in the experiment without preprocessing. The sizable improvement in the results shows that even simple data processing can enhance the model.

Table 4
Results for Task 2A: Detecting Previously Fact-Checked Claims in Tweets on the test dataset. *DPP stands for *Data Preprocessing*.

EXPERIMENT	MRR	MAP1	MAP3	MAP5	MAP10	MAP20	P1	P3	P5	P10	P20
BM25: *Vclaim*	.752	.688	.733	.743	.749	.751	.688	.262	.165	.087	.045
BM25: *Vclaim* + *Title*	.761	.703	.741	.749	.757	.759	.703	.262	.165	.087	.045
BM25: *Vclaim* + *Title* + *Subtitle* + *Date*	.744	.688	.725	.734	.739	.742	.688	.257	.162	.085	.044
BM25: *Vclaim* (+ DPP*)	.772	.708	.753	.761	.769	.771	.708	.271	.169	.090	.046
BM25: *Vclaim* + *Title* (+ DPP)	.788	.727	.772	.777	.785	.787	.728	.277	.170	.091	.047
BM25: *Vclaim* + *Title* + *Subtitle* + *Date* (+ DPP)	.815	.738	.804	.809	.812	.814	.738	.294	.180	.092	.047
S-BERT (+ DPP)	.795	.728	.778	.787	.791	.794	.728	.282	.177	.092	.048
RankSVM: Top-20	.896	.866	.890	.896	.896	.896	.866	.307	.189	.094	.047
RankSVM: Top-50	**.909**	**.876**	**.903**	**.908**	**.909**	**.909**	**.876**	.313	.193	.097	**.049**
RankSVM: Top-100	.908	.866	.902	**.908**	**.909**	**.909**	.866	**.317**	**.195**	**.098**	**.049**

5.3.2. S-BERT

This experiment follows the setup described in Section 4.3. Due to BM25 achieving better performance for preprocessed data, we decided to train the classifier over the preprocessed training dataset. The fact-checked claims are constructed of all semantically meaningful fields for Task 2A - *Title*, *Subtitle*, *Vclaim* and *Date*. The performance of the trained classifier over the preprocessed test data is reported in Table 4. We can see that S-BERT achieves a lower MRR of 0.795, compared to the best BM25 result, which is 0.815. However, the results are relatively close, and thus we conclude that for Task 2A, in order to fact-check the input claim, both semantic and exact matching between the input claims and the fact-checked claims should be used.

5.3.3. RankSVM

Using RankSVM, we re-ranked the retrieved results from *BM25: Vclaim + Title + Subtitle + Date (+ DPP)*, because they yielded the best performance. The training was performed as described in Section 4.4. The optimal value for N was chosen empirically after experiments with N equal to 20, 50, and 100. Table 4 shows that the best performance is achieved for N = 50, and after a certain point, as the length of the re-ranking list increases, the result is getting worse (for N = 100). Using RankSVM for re-ranking makes sense because it notably improves the results, compared to the best individual model, MRR increases from 0.815 to 0.909 for top-50, which leads to almost 10% improvement.

5.4. Results Task 2B (Debates)

5.4.1. BM25

For Task 2B, we calculate scores for exact matching between the input claim and the fields *Vclaim*, *Text*, *Title* + *Vclaim* + *Text* and *Title* + *Vclaim* + *Date* + *Text*. Similarly to Task 2A, the results are for both preprocessed and non-preprocessed data.

In Table 5, we can see that for non-preprocessed data, the *Text* field itself yields better results than *Vclaim*: MRR for *Vclaim* is 0.360, and it increases to 0.415 for *Text*. The data analysis described in Section 3.3 shows that the *Text* is longer than *Vclaim*, and thus it has higher probability of containing information related to the input claim.

Table 5
Results for Task 2B: Detecting Previously Fact-Checked Claims in Political Debates/Speeches on the test dataset. *DPP stands for *Data Preprocessing*.

EXPERIMENT	MRR	MAP1	MAP3	MAP5	MAP10	MAP20	P1	P3	P5	P10	P20
BM25: *Vclaim*	.360	.316	.355	.359	.316	.362	.316	.152	.096	.052	.028
BM25: *Text*	.415	.348	.393	.401	.408	.409	.342	.181	.118	.064	.034
BM25: *Title* + *Vclaim* + *Text*	.422	.367	.396	.406	.413	.416	.367	.168	.111	.062	.033
BM25: *Title* + *Vclaim* + *Date* + *Text*	.390	.342	.360	.376	.382	.386	.348	.152	.109	.059	.034
BM25: *Vclaim* (+ DPP*)	.363	.316	.359	.362	0.365	0.368	.316	.156	.096	.052	.028
BM25: *Text* (+ DPP)	.412	.348	.391	.395	.402	.405	.342	.177	.114	.062	.034
BM25: *Title* + *Vclaim* + *Text* (+ DPP)	.424	.316	.359	.362	.365	.369	.316	.156	.096	.052	.028
BM25: *Title* + *Vclaim* + *Date* + *Text* (+ DPP)	.399	.361	.376	.389	.340	.403	.354	.152	.106	.062	.034
S-BERT (+ DPP)	.336	.278	.313	.328	.338	.342	.266	.143	.099	.059	.032
RankSVM: Top20	.488	.443	.487	.489	.492	.493	.443	.207	.127	.066	.034
RankSVM: Top50	.500	.456	.498	.502	.508	.509	.456	.211	.134	.072	.037
RankSVM: Top100	.497	.443	.482	.497	.504	.505	.443	.203	.137	.075	.038
RankSVM: Top20 + *Date*	.508	.468	.502	.506	.506	.506	.481	.207	.134	.067	.033
RankSVM: Top50 + *Date*	.533	.487	.531	.535	.536	.536	.493	.227	.144	.073	.037
RankSVM: Top100 + *Date*	**.559**	**.494**	**.546**	**.551**	**.557**	**.558**	**.506**	**.232**	**.149**	**.080**	**.040**
RankSVM: Top150 + *Date*	.533	.487	.531	.535	.536	.536	.494	.224	.144	.073	.037

Moreover, we notice that combining the three fields *Title*, *Vclaim* and *Text* yields the best representation: MRR of 0.422. This improves the result by 0.01–0.06 points compared to the MRR when using just a single field. We expect that this is because combining all fields provides more detail about the fact-checked claim. On the other hand, adding the *Date* field to *Title* + *Vclaim* + *Text* degrades MRR by 0.032, because of the increase of redundant information.

Then, we applied data preprocessing. Table 5 shows that the results for the *Text* field are slightly worse: MRR for the experiment with no preprocessing decreases from 0.415 to 0.412. On the other hand, we notice an improvement for *Text* and *Title* + *Vclaim* + *Text* by 0.02–0.04 points in terms of MRR and MAP@5. We tried to add the *Date* in the same way as in Task 2A, but with no effect, probably because the *Text* field in Task 2B consists of many sentences and the last sentence may not be similar to the *Iclaim* at all. Moreovers, we added the *Date* as part of the last sentence of the *Vclaim*, and we combined it with all three fields, *Title* + *Vclaim* + *Text*, but the MRR decreased by 0.02 compared to using only the three fields. We believe that the almost insignificant improvement of data preprocessing is due to the fact that removing new lines only shortens the text without affecting the exact matching of the claims. Furthermore, the transcripts from political debates and speeches are usually formally written and the probability of them containing sequences that have to be split into strings is low.

5.4.2. S-BERT

We trained S-BERT on preprocessed data and on the combination of the fields *Title*, *Vclaim*, and *Text*, as this yielded the best performance for BM25. In Table 5, we can see that the S-BERT performs much worse than BM25. Also, S-BERT decreases the value of MRR from 0.424 to 0.336. We believe that the lower result for S-BERT is because we calculate the cosine similarity between the input claim and each sentence of the article discussing the previously fact-checked claim, and the final score is based on the top-4 sentences. If these top-4 sentences do not contain enough context, the semantic similarity will be low. On the other hand, we using scores from more than four sentences would add noise and hurt more than help.

5.4.3. RankSVM

For Task 2B, we used the BM25 experiment described in Section 5.4.1 as a starting point for re-ranking, as it yielded the best results. The training process is presented in Section 4.4. We can see in Table 5 that RankSVM enhances the best individual model by increasing MRR from 0.424 to 0.500. The best results are achieved for $N = 50$, and they start degrading for longer lists.

Moreover, after the unsuccessful attempt to add the *Date* field to BM25, we decided to include it as a feature in RankSVM. Section 4.4 describes how the feature is extracted. Table 5 shows consistent improvement over RankSVM:Top50 without the date feature, by 0.05 in terms of MRR and by 0.03-0.05 in terms of MAP@k. The best result is achieved for $N = 100$.

Using RankSVM significantly improves the results for both tasks. We notice that the best re-ranking model for Task 2A is when using a top-50 list, compared to top-100 for Task 2B. This observation confirms the assumption in [4] that the optimal list length is dependent on the different performance of the retrieval models used to extract the top-N pairs. For Task 2A, *BM25: Vclaim + Title + Subtitle + Date* has MRR of 0.815, and for Task 2B, *BM25:Vclaim + Title + Text* has MRR of 0.424. Hence, the BM25 model, which is being re-ranked for Task 2A, is stronger and there is no need to search for relevant fact-checked claims in a longer list.

6. Conclusion and Future Work

Our findings indicate that the combination of proper data handling, along with the BM25 algorithm, S-BERT embeddings, and neural networks, can yield a simple and fast mechanism for fact-checked claim detection with high performance. We use a re-ranking algorithm based on RankSVM to capture the different types of information obtained from BM25 and S-BERT. However, despite the fact that we managed to improve our system, so that it now performs better than all of the proposed solutions as part of the competition (including ours), our approach still needs fine-tuning before it can be reliably used as a self-sufficient tool for verification whether a given claim has been previously fact-checked.

In future work, we plan enhancements of the proposed system using data augmentation with existing datasets for fact extraction and verification, such as FEVER, adding named entity recognition as a feature to the re-ranker, and adding information from the URLs in the tweets. Moreover, we plan to continue our research over data augmentation using back-translation.

Acknowledgments

This research was conducted in the frame of Knowledge Discovery courses at the faculty of Mathematics and Informatics, Sofia University “St. Kliment Ohridski”. We want to thank Prof. Ivan Koychev for his support and advise. This research is partially supported by the National Scientific Program “ICTinSES”, financed by the Bulgarian Ministry of Education and Science.

This work is also part of the Tanbih mega-project, (http://tanbih.qcri.org) developed at the Qatar Computing Research Institute, HBKU, which aims to limit the impact of “fake news”, propaganda, and media bias by making users aware of what they are reading, thus promoting media literacy and critical thinking.

References

[1] F. Alam, F. Dalvi, S. Shaar, N. Durrani, H. Mubarak, A. Nikolov, G. D. S. Martino, A. Abdelali, H. Sajjad, K. Darwish, P. Nakov, Fighting the COVID-19 infodemic in social media: A holistic perspective and a call to arms, in: Proceedings of the Fifteenth International AAAI Conference on Web and Social Media, ICWSM '21, 2021, pp. 913–922.

[2] A. Zubiaga, M. Liakata, R. Procter, G. Wong Sak Hoi, P. Tolmie, Analysing how people orient to and spread rumours in social media by looking at conversational threads, PloS one 11 (2016) e0150989.

[3] S. Shaar, F. Haouari, W. Mansour, M. Hasanain, N. Babulkov, F. Alam, G. Da San Martino, T. Elsayed, P. Nakov, Overview of the CLEF-2021 CheckThat! lab task 2 on detect previously fact-checked claims in tweets and political debates, in: Working Notes of CLEF 2021—Conference and Labs of the Evaluation Forum, CLEF '2021, Bucharest, Romania (online), 2021.

[4] S. Shaar, N. Babulkov, G. Da San Martino, P. Nakov, That is a known lie: Detecting previously fact-checked claims, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL '20, 2020, pp. 3607–3618.

[5] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP '19, Hong Kong, China, 2019, pp. 3982–3992.

[6] T. Joachims, Optimizing search engines using clickthrough data, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02, Edmonton, Alberta, Canada, 2002, pp. 133–142.

[7] M. Bouziane, H. Perrin, A. Cluzeau, J. Mardas, A. Sadeq, Team Buster.ai at CheckThat! 2020 insights and recommendations to improve fact-checking, in: CLEF, 2020.

[8] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, R. Procter, Detection and resolution of rumours in social media: A survey, ACM Comput. Surv. 51 (2018).

[9] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, J. Han, A survey on truth discovery, SIGKDD Explor. Newsl. 17 (2016) 1–16.

[10] G. D. S. Martino, S. Cresci, A. Barrón-Cedeño, S. Yu, R. D. Pietro, P. Nakov, A survey on computational propaganda detection, in: C. Bessiere (Ed.), Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI '20, 2020, pp. 4826–4832.

[11] M. Hardalov, A. Arora, P. Nakov, I. Augenstein, A survey on stance detection for mis- and disinformation identification, arXiv preprint arXiv:2103.00242 (2021).

[12] A. Barrón-Cedeño, T. Elsayed, P. Nakov, G. Da San Martino, M. Hasanain, R. Suwaileh, F. Haouari, N. Babulkov, B. Hamdan, A. Nikolov, S. Shaar, Z. Ali, Overview of CheckThat! 2020 — automatic identification and verification of claims in social media, in: Proceedings of the 11th International Conference of the CLEF Association: Experimental IR Meets Multilinguality, Multimodality, and Interaction, CLEF '2020, Thessaloniki, Greece, 2020, pp. 215–236.

[13] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human

Language Technologies, NAACL-HLT '19, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[14] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, H. Hajishirzi, Fact or fiction: Verifying scientific claims, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP '20, 2020, pp. 7534–7550.

[15] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, FEVER: a large-scale dataset for fact extraction and VERification, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT '18, New Orleans, Louisiana, 2018, pp. 809–819.

[16] W. Y. Wang, "Liar, liar pants on fire": A new benchmark dataset for fake news detection, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL '17, Vancouver, Canada, 2017, pp. 422–426.

[17] I. Ounis, G. Amati, V. Plachouras, B. He, C. MacDonald, C. Lioma, Terrier : A high performance and scalable information retrieval platform, 2006.

[18] J. Tiedemann, S. Thottingal, OPUS-MT – building open translation services for the world, in: Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, Lisboa, Portugal, 2020, pp. 479–480.

[19] 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. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP '20, 2020, pp. 38–45.

[20] N. Craswell, Mean Reciprocal Rank, Springer US, Boston, MA, 2009, pp. 1703–1703.