Team_Alex at CLEF CheckThat! 2020: Identifying Check-Worthy Tweets With Transformer Models

Alex Nikolov[1], Giovanni Da San Martino[2], Ivan Koychev[1], and Preslav Nakov[2]

[1] FMI, Sofia University “St Kliment Ohridski”, Bulgaria
[2] Qatar Computing Research Institute, HBKU, Qatar

Abstract. While misinformation and disinformation have been thriving in social media for years, with the emergence of the COVID-19 pandemic, the political and the health misinformation merged, thus elevating the problem to a whole new level and giving rise to the first global infodemic. The fight against this infodemic has many aspects, with fact-checking and debunking false and misleading claims being among the most important ones. Unfortunately, manual fact-checking is time-consuming and automatic fact-checking is resource-intense, which means that we need to pre-filter the input social media posts and to throw out those that do not appear to be check-worthy. With this in mind, here we propose a model for detecting check-worthy tweets about COVID-19, which combines deep contextualized text representations with modeling the social context of the tweet. Our official submission to the English version of CLEF-2020 CheckThat! Task 1, system Team_Alex, was ranked second with a MAP score of 0.8034, which is almost tied with the wining system, lagging behind by just 0.003 MAP points absolute.

1 Introduction

The rise of disinformation (aka “fake news”) in social media has given rise to a number of initiatives to fact-check claims of general interest and to confirm or to debunk them. Unfortunately, manual fact-checking is a very time-consuming process, and thus automated approaches have been proposed as a faster alternative. Yet, even with automated methods, it is not possible to fact-check every single claim, and the accuracy of automated fact-checking systems is significantly lower than that of human experts. Furthermore, there is a need to pre-filter and to prioritize what should be passed to human fact-checkers. As the need to prioritize has been gradually gaining recognition, so was the task of check-worthiness estimation, which is seen as an important first step in the general fact-checking pipeline. A leading effort in this direction has been the CLEF CheckThat! lab, which featured a check-worthiness estimation task in all its editions [1,2,18].

Traditionally, check-worthiness estimation has focused on political debates and speeches, ignoring social media. In order to bridge this gap, the 2020 edition of the CLEF-2020 CheckThat! Lab [3] featured Task 1 on check-worthiness estimation on tweets, offered in Arabic and English [10,18]. Given the prominence of disinformation related to COVID-19, which has grown to become the first global infodemic, the English Task 1 focused on tweets related to COVID-19.

The lab organizers provided a dataset of tweets originating from the early days of the global COVID-19 pandemic and covering a variety of COVID-19-related topics, e.g., concerning the number of confirmed cases in different parts of the world, the measures taken by local governments to combat the pandemic, claims about the nature of the virus, etc. The participants were challenged to develop systems to rank a set of input tweets according to their check-worthiness.

Below, we describe the system we developed for the English Task 1, which is an ensemble combining deep contextualized text representations with social context. Our official submission was ranked second-best, and it was almost tied with the winner. We further describe a number of additional experiments and comparisons, which we believe should be useful for future research as they provide some indication about what techniques are effective for the task.

2 Related Work

The earliest work on claim check-worthiness estimation is the ClaimBuster system [11], which was trained on manually annotated US Presidential debates. It used TF.IDF features after discarding words appearing in less than three documents. In addition, for each sentence they calculated a sentiment score, word counts, number of occurrences for 43 Part-of-Speech (POS) tags from the Penn Treebank, as well as the frequency of use of 26 entity types, such as *Person* and *Organization*. They performed feature selection using a random forest and GINI index, and then conducted various experiments using feature subsets passed on to a Support Vector Machine, a Random Forest, and a Naïve Bayes classifiers.

In a related line of work, Gencheva & al. [9] created a dataset also based on US Presidential debates, but with annotations from nine fact-checking organizations. Their goal was to mimic the selection strategies of these organizations, and they focused on modeling the context. They reused most of ClaimBuster's features and added the number of named entities within a sentence, the number of words belonging to one of nine possible lexicons, i.e., words indicating bias or negative words. For the context, they added features for sentence positioning within a speaker's segment, for the size of the segment a sentence belongs to, as well as for the size of the previous and of the next segments, along with several metadata features. They further trained an LDA topic model [4] on Presidential debates and used it to extract a distribution over 300 learned topics, which they used as additional features. Finally, they added averaged word2vec word embeddings [14] for each sentence. They trained an SVM classifier as well as a feed-forward neural network with two hidden layers, and ReLU for activation, and they found that the additional context features yielded sizable performance gains.

In follow-up work, Vasileva & al. [19] used a multi-task learning neural network that predicts whether a sentence would be selected for fact-checking by each individual fact-checking organization (from a set of nine such organizations), as well as by any of them.

Yet another follow-up work resulted in the development of the ClaimRank system, which was trained on more data and also included Arabic content [12]. Other related work, also focused on political debates and speeches. For example, Patwari & al. [16] predicted whether a sentence would be selected by a fact-checking organization using a boosting-like model.

Last but not least, the task was the topic of CLEF in 2018 [1,15] and 2019 [2,7,8], where the focus was once again on political debates and speeches, from a single fact-checking organization.

In a slightly different domain, Konstantinovskiy & al. [13] developed a dataset for check-worthiness estimation consisting of manually annotated sentences from TV debates in the UK. They used InferSent sentence embeddings [6], as well as the number of occurrences of some POS tags, and the number of different named entities within a sentence. They experimented with a number of classifiers such as Logistic Regression, SVMs, Naïve Bayes, and Random Forest.

Note that all above systems were trained on speeches and debates, while the task we deal with here is about tweets. While we reuse some of the features of these systems, we focus on the first step for finding appropriate data representation, i.e., using pre-processing techniques specific for tweets.

3 Data Pre-processing

We applied different pre-processing techniques to the raw tweet text, which we describe in the following subsections.

3.1 Default Pre-processing

We will begin with the description of our *default* pre-processing. It includes the following processing rules and heuristics:

Splitting hashtags into separate words based on UpperCamelCase. The UpperCamelCase convention is a way to write multiple words joined together as a single word with the first letter of each of the multiple words capitalized within the joined word. It is a technique commonly used in hashtags in Twitter, e.g., the string `#TheMoreYouKnow` is composed of four words: *the*, *more*, *you*, and *know*. Such joined-words hashtags are common in Twitter, and thus we attempted to split them, possibly extracting useful text and facilitating the understanding of the tweets. As hashtags can deviate from the standard UpperCamelCase convention, we further added some additional rules to cope with some variations.

Unification of the hashtags about COVID-19. The dataset contains tweets with hashtags such as #covid19, #covid_19, #Covid2019. We replaced all such hashtags with the canonical tag #covid-19. Similarly, we unified the different ways of spelling (or even misspelling) the colloquial term *corona virus*, including hashtags such as #coronavirus, #corona, #korona; we replaced all such wordforms and hashtags with the canonical form *corona virus*.

Replacing '@' with 'user'. In general, the identity of a tweet's author is irrelevant regarding a tweet's check-worthiness, and thus we replaced all user mentions with the special token *user*. However, we preserved the identities of influential public figures and organizations since their status might influence the check-worthiness of the respective tweets. In such cases, we replaced certain usernames with the person's actual name or title. For example, we replaced @realDonaldTrump with *Donald Trump*, @VP with *Vice President*, and @WHO with *World Health Organization*.

Replacing URLs with 'url'. The presence of a URL can have an impact on the final check-worthiness label. A classifier might have difficulties differentiating between different target URLs, and thus we replaced all URLs by the special *url* token.

Removing hashtags at the end of tweets. Many tweets contained hashtags coming after the meaningful textual statement, and possibly before an included URL, e.g., *'This is a scandal! #Covid-19 #Upset #Scandal'*. We observed that such hashtags typically did not bring much information with respect to check-worthiness, and thus we removed them.

Expanding shortened quantities. We replaced tokens such as *7m* and *12k* with expanded versions such as *7 million* and *12 thousand*, respectively.

Removing punctuation marks, except for quotation marks. Punctuation does not help much semantically, and thus we removed it. However, we kept quotation marks, which can often indicate the beginning or the ending of a quote, which may be the key point of a claim worth fact-checking.

3.2 Corona Pre-processing

We further applied a special pre-processing hack, which we call *Corona* pre-processing. It replaces the words *covid-19* and *corona virus* with *ebola*, which helps to obtain a more meaningful semantic representation of the input text, as *ebola* is in the vocabulary of pre-trained embeddings and Transformers, while *covid-19* and *corona virus*, which are more recent terms, are not.

3.3 SW+C Pre-processing

A third method of pre-processing, which we call *SW+C*, applies the rules of the *Corona* pre-processing, and further removes most stop-words (as listed in the standard NLTK stop-word list). However, it keeps personal and demonstrative pronouns, such as *he* and *this*, as they provide references that might be important to determine whether a claim is check-worthy.

4 Experiments

We experimented with the above pre-processing techniques and different representations and learning models. All results reported in Tables 1-3 show the average performance using 5-fold cross-validation on the validation set.

Our baseline system is an SVM with TF.IDF tokens as features, and it achieved a MAP score of 0.6235.

4.1 Experiments with SVM

We used an SVM classifier with word embeddings from GloVe, FastText, Sent2vec, as well as from Transformers. For GloVe and for FastText, we used three different pooling options: mean pooling, max pooling, and TF-IDF pooling. For embeddings from Transformers, we used mean pooling, max pooling, direct use of the CLS token, and WK pooling [20].

For each embedding type and pre-processing technique, we performed randomized search with 1,000 iterations. The space of search hyper-parameters consisted of three different kernel types: linear, polynomial, and RBF, each with an equal probability of being selected. In addition, the regularization parameter C and the kernel coefficient for the RBF and for the polynomial kernels γ were sampled from a Gamma distribution with parameters α=2, β=1, so that most of our samples are close to the default value of 1 in order to prevent overfitting; however, in some cases, extreme values were also tried. When selecting a polynomial kernel, a degree parameter also needs to be provided. We sampled the degree uniformly with values between 2 and 5. The results of the experiments are shown in Table 1. In addition to reporting the highest achieved MAP score for a given group, we further show the corresponding macro-F1 score.

We can see in Table 1 that using mean pooling alongside GloVe embeddings yields marginally better results than using max pooling or TF-IDF pooling. The same is true for the experiments with FastText embeddings, which achieved a slightly higher MAP and macro-F1 scores. The Twitter bigrams Sent2Vec embeddings, combined with a pre-processing of replacing all occurrences of *Covid-19* and *Corona virus* with *Ebola*, managed to outperform the GloVe and the FastText embeddings. Finally, using Transformer embeddings yielded the best overall MAP and macro-F1 scores; note that these results were achieved using WK pooling on the BERT-base embeddings.

Table 1. SVM experiments: Shown is the highest dev MAP and the corresponding macro-F1 score. The best results for each pre-processing and evaluation measure are underlined, while the best overall result is marked in bold.

	Corona		**SW+C**	
Embedding+Pooling	**MAP**	**macro-F1**	**MAP**	**macro-F1**
GloVe mean pooling	0.7140	0.6989	0.7239	0.7168
GloVe max pooling	0.6709	0.6876	0.6632	0.6510
GloVe TF-IDF pooling	0.6938	0.7074	0.6974	0.7057
FastText mean pooling	0.7151	0.7104	0.7286	0.7300
FastText max pooling	0.6852	0.6968	0.6801	0.6972
FastText TF-IDF pooling	0.6909	0.6932	0.6923	0.6985
Wiki unigrams	0.7020	0.6948	0.6563	0.6239
Wiki bigrams	0.6981	0.3929	0.6883	0.6937
Twitter unigrams	0.7133	0.7084	0.7092	0.7121
Twitter bigrams	0.7315	0.7135	0.7141	0.7266
Toronto unigrams	0.6589	0.6877	0.6473	0.6808
Toronto bigrams	0.6795	0.6966	0.6767	0.6865
BERT base	0.7454	0.7300	0.7535	**0.7656**
BERT large	0.7015	0.7228	0.7055	0.7098
DistilBERT	0.7357	0.7394	0.7317	0.7022
RoBERTa base	0.7178	0.6877	0.7332	0.7373
RoBERTa large	0.7164	0.7349	0.7254	0.7256

4.2 Experiments with Logistic Regression

We experimented with Logistic Regression in a setup, similar to that for SVM: we performed randomized search with 1,250 iterations and 5-fold cross-validation. The hyper-parameter search space includes different solvers such as Newton-cg, SAG, LBFGS, SAGA, and Liblinear, and the regularization parameter C was sampled from a Gamma distribution with parameters α=2, β=1.

Table 2 shows the evaluation results. The highest MAP score for GloVe embeddings is achieved with mean pooling, similarly to SVM. In contrast, the best MAP scores on FastText embeddings are achieved using max pooling (whereas mean pooling was best with SVM). Overall, the results with Logistic Regression and Glove & FastText embeddings are moderately lower than those with SVM.

The Sent2Vec embedding experiments yielded the best results with Twitter bigram embeddings (as was the case with SVM). However, they only yielded a tiny improvement over the highest scores using GloVe and FastText embeddings.

The results using Transformer embeddings as features for Logistic Regression are similar to the ones with SVM: in both cases, WK pooling with BERT-base worked best, and the MAP scores were also similar. In the case of Logistic Regression, the use of Transformer embeddings considerably improved the MAP scores compared to the use of GloVe, FastText, and Sent2Vec embeddings.

Table 2. Logistic regression experiments: Shown are the highest dev MAP and the corresponding macro-F1 score. The best results for each pre-processing and evaluation measure are underlined, while the best overall result is marked in bold.

	Corona		**SW+C**	
Embedding+Pooling	**MAP**	**macro-F1**	**MAP**	**macro-F1**
GloVe mean pooling	0.6569	0.6755	0.6714	0.6781
GloVe max pooling	0.6627	0.6783	0.6583	0.6703
GloVe TF-IDF pooling	0.6460	0.6653	0.6502	0.6648
FastText mean pooling	0.6536	0.6937	0.6592	0.6759
FastText max pooling	0.6638	0.6594	0.6567	0.6664
FastText TF-IDF pooling	0.6431	0.6691	0.6459	0.6715
Wiki unigrams	0.6554	0.6871	0.6414	0.6856
Wiki bigrams	0.6617	0.6601	0.6525	0.6567
Twitter unigrams	0.6712	0.6748	0.6520	0.6593
Twitter bigrams	0.6741	0.6739	0.6649	0.6661
Toronto unigrams	0.6408	0.6540	0.6396	0.6530
Toronto bigrams	0.6429	0.6448	0.6422	0.6335
BERT base	0.7494	0.7433	0.7510	**0.7539**
BERT large	0.7255	0.7325	0.7134	0.7195
DistilBERT	0.7140	0.7053	0.7156	0.7018
RoBERTa base	0.7488	0.7300	0.7382	0.7381
RoBERTa large	0.7002	0.7201	0.7002	0.7201

4.3 Experiments with Transformers

For Transformers, we used randomized search with 5-fold cross-validation for 20 iterations. We sampled the hyper-parameters uniformly as follows: the number of training epochs from $\{2, 3, 4, 5\}$, the batch size from $\{2, 4, 8, 12, 16, 24, 32\}$, and the learning parameter from $\{6.25e - 5, 5e - 5, 3e - 5, 2e - 5\}$.

The results are shown in Table 3. Among the Transformer models, RoBERTa achieved the highest MAP score by a wide margin and for all pre-processing techniques. The highest score was achieved using the *Corona* pre-processing, which is the best overall result. The hyper-parameters that performed the best used 5 training epochs, a batch size of 8, and a learning rate of 3e-05.

4.4 Experiments using Tweet Metadata

Finally, we used information about the tweet and its author, e.g., the number of retweets of the target tweet, the number of friends the tweet's author has, the number of years since the account was created, the presence of a URL in the tweet, etc. We used these extra features by concatenating them to the validation set predictions for the best-performing RoBERTa models, which arose from the 5-fold cross-validation. The best set of parameters for each RoBERTa model and pre-processing is shown in Table 4.

Table 3. Transformer experiments: Shown is the highest dev MAP and the corresponding macro-F1 score using Logistic Regression, embeddings from Transformers, and different pooling techniques. The best results for each pre-processing and evaluation measure are underlined, while the best overall result is marked in bold.

	Corona		**SW+C**		**Default**	
Embedding+Pooling	**MAP**	**macro-F1**	**MAP**	**macro-F1**	**MAP**	**macro-F1**
BERT base	0.7384	0.7255	0.7311	0.7523	0.7371	0.7558
RoBERTa base	0.7860	0.7672	0.7552	0.7492	0.7854	**0.7880**
DistilBERT	0.7433	0.7297	0.7405	0.7411	0.7476	0.7442
AlBERT base	0.7340	0.7140	0.6802	0.6471	0.6625	0.6376

As some tweets contain a link to online news articles, we designed features modeling the factuality of reporting of the outlets that published these news articles. For this, we used the manual judgments from Media Bias/Fact Check.[3]

We thus derived the following nine Boolean features:

- Is Twitter account verified?
- Does the tweet contain a URL?
- Does the tweet contain a link to an article published by a news outlet whose factuality of reporting is
  - very high?
  - high?
  - mostly factual?
  - mixed?
  - low?
  - fake news?
  - conspiracy?

As well as the following three numerical features:

- Natural logarithm of the number of times the tweet was retweeted;
- Natural logarithm of the number of friends of the Twitter account;
- Years since the registration of the Twitter account.

We conducted experiments concatenating the RoBERTa predictions to the first nine and also to all twelve metadata features. The results are shown in Tables 5 and 6, respectively. Once again, we ran randomized search using SVM and Logistic Regression on the new features, while searching in the same hyperparameter space as described in Sections 4.1 and 4.2. We can see that Logistic Regression performed better than SVM. Moreover, while the best MAP scores are identical when using twelve vs. nine metadata features, using all twelve features performed a bit better in terms of macro-F1 score.

[3] `http://mediabiasfactcheck.com`

Table 4. Metadata experiments: Pre-processing and hyper-parameters for the best-performing RoBERTa models on the validation set.

Pre-processing	**Training Epochs**	**Batch Size**	**Learning Rate**	**MAP**
Default	5	8	3.00e-05	0.7900
Corona	4	2	2.00e-05	0.7875
Default	4	8	3.00e-05	0.8854
Corona	4	24	5.00e-05	0.7442
Corona	5	8	3.00e-05	0.8363

Table 5. Metadata experiments: Highest MAP and corresponding macro-F1 scores for SVM and Logistic Regression using the RoBERTa prediction and the first nine metadata features.

Model	**MAP**	**macro-F1**
SVM	0.7994	0.7893
Logistic Regression	0.8017	0.7868

Table 6. Metadata experiments: Highest MAP and corresponding macro-F1 scores for SVM and Logistic Regression using the RoBERTa prediction and all twelve metadata features.

	MAP	**macro-F1**
SVM	0.7993	0.7893
Logistic Regression	0.8017	0.7864

5 Official Results on the Test Set

For our primary submission, we used Logistic Regression with all twelve metadata features in addition to RoBERTa predictions, as described in Section 4.4.

For our first and second contrastive runs, we used models based solely on RoBERTa. Our contrastive 2 run averages the test set predictions of each of the trained RoBERTa models with identical hyper-parameters, whereas our contrastive 1 run averages the predictions of the best-performing RoBERTa models on the corresponding validation sets. For the contrastive 1 run, we did not pose the restriction of having identical hyper-parameters.

The official evaluation results are shown in Table 7. We can see that our primary run achieved a MAP score of 0.8034, which puts us at second place, falling behind the winner by 0.003 points absolute only, i.e., we are practically tied for the first place. Our primary run achieved a higher MAP score on the test set than on the validation sets, which is a sign of a model that is capable of generalizing well and not overfitting. Our first and second contrastive runs also achieved high MAP scores on the test set, but were slightly worse.

Table 7. Official results on the test set. Shown are the results for our primary and constrastive submissions.

Runs	**MAP**	**R-Pr**	**P@1**	**P@3**	**P@5**	**P@10**	**P@20**	**P@30**
Primary	0.8034	0.6500	1.0000	1.0000	1.0000	1.0000	0.9500	0.7400
Contrastive 1	0.7988	0.6500	1.0000	1.0000	1.0000	1.0000	0.9500	0.7400
Contrastive 2	0.7809	0.6667	1.0000	1.0000	1.0000	1.0000	0.8500	0.6800

6 Conclusion and Future Work

We have described our system, Team_Alex, for detecting check-worthy tweets about COVID-19, which we developed for the English version of CLEF-2020 CheckThat! Task 1. It is based on an ensemble combining deep contextualized text representations with social context, as well as advanced pre-processing. Our system was ranked second with a MAP score of 0.8034, which is almost tied with the wining system, lagging behind by just 0.003 MAP points absolute.

We further described a number of additional experiments and comparisons, which we believe should be useful for future research as they provide some indication about what techniques are effective for the task.

In future work, we plan to experiment with more pre-processing techniques, with better modeling the social context, as well as with some newer Transformer models such as T5 [17] and Electra [5].

Acknowledgments

This research is part of the Tanbih project,[4] developed at the Qatar Computing Research Institute, HBKU, which aims to limit the effect of “fake news”, propaganda, and media bias by making users aware of what they are reading, thus promoting media literacy and critical thinking.

This research is also partially supported by the National Scientific Program “ICTinSES”, financed by the Bulgarian Ministry of Education and Science and Project UNITe BG05M2OP001-1.001-0004, funded by the OP “Science and Education for Smart Growth” and the EU via the ESI Funds.

[4] `http://tanbih.qcri.org`

References

[1] Atanasova, P., Màrquez, L., Barrón-Cedeño, A., Elsayed, T., Suwaileh, R., Zaghouani, W., Kyuchukov, S., Da San Martino, G., Nakov, P.: Overview of the CLEF-2018 CheckThat! Lab on automatic identification and verification of political claims, Task 1: Check-worthiness. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) CLEF 2018 Working Notes. Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, CEUR-WS.org, Avignon, France (2018)

[2] Atanasova, P., Nakov, P., Karadzhov, G., Mohtarami, M., Martino, G.D.S.: Overview of the CLEF-2019 CheckThat! lab on automatic identification and verification of claims. Task 1: Check-worthiness. In: Cappellato, L., Ferro, N., Losada, D., Müller, H. (eds.) CLEF 2019 Working Notes. Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, CEUR-WS.org, Lugano, Switzerland (2019)

[3] Barrón-Cedeño, A., Elsayed, T., Nakov, P., Da San Martino, G., Hasanain, M., Suwaileh, R., Haouari, F., Babulkov, N., Hamdan, B., Nikolov, A., Shaar, S., Ali, Z.: 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)

[4] Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. Journal of Machine Learning Research **3**, 601–608 (01 2001)

[5] Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: ELECTRA: Pre-training text encoders as discriminators rather than generators. arXiv pre-print 2003.10555 (2020)

[6] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. arXiv pre-print 1705.02364 (2017)

[7] Elsayed, T., Nakov, P., Barrón-Cedeño, A., Hasanain, M., Suwaileh, R., Atanasova, P., Da San Martino, G.: CheckThat! at CLEF 2019: Automatic identification and verification of claims. In: Proceedings of the 41st European Conference on Information Retrieval. pp. 309–315. ECIR '19, Cologne, Germany (2019)

[8] Elsayed, T., Nakov, P., Barrón-Cedeño, A., Hasanain, M., Suwaileh, R., Da San Martino, G., Atanasova, P.: Overview of the CLEF-2019 CheckThat!: Automatic identification and verification of claims. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. LNCS, Springer, Lugano, Switzerland (2019)

[9] Gencheva, P., Nakov, P., Màrquez, L., Barrón-Cedeño, A., Koychev, I.: A context-aware approach for detecting worth-checking claims in political debates. In: Proceedings of the Conference Recent Advances in Natural Language Processing. pp. 267–276. RANLP '17, Varna, Bulgaria (2017)

[10] Hasanain, M., Haouari, F., Suwaileh, R., Ali, Z., Hamdan, B., Elsayed, T., Barrón-Cedeño, A., Da San Martino, G., Nakov, P.: Overview of CheckThat! Arabic: Automatic identification and verification of claims in social media. In: Working Notes of CLEF 2020—Conference and Labs of the Evaluation Forum. CLEF '2020, Thessaloniki, Greece (2020)

[11] Hassan, N., Li, C., Tremayne, M.: Detecting check-worthy factual claims in presidential debates. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. p. 1835–1838. CIKM '15, Melbourne, Australia (2015)

[12] Jaradat, I., Gencheva, P., Barrón-Cedeño, A., Màrquez, L., Nakov, P.: ClaimRank: Detecting check-worthy claims in Arabic and English. In: Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 26–30. NAACL-HLT '18, New Orleans, Louisiana, USA (2018)

[13] Konstantinovskiy, L., Price, O., Babakar, M., Zubiaga, A.: Towards Automated Factchecking: Developing an Annotation Schema and Benchmark for Consistent Automated Claim Detection. arXiv pre-print 1809.08193 (2018)

[14] Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting Similarities among Languages for Machine Translation. arXiv e-prints arXiv:1309.4168 (2013)

[15] Nakov, P., Barrón-Cedeño, A., Elsayed, T., Suwaileh, R., Màrquez, L., Zaghouani, W., Atanasova, P., Kyuchukov, S., Da San Martino, G.: Overview of the CLEF-2018 CheckThat! Lab on automatic identification and verification of political claims. In: Proceedings of the Ninth International Conference of the CLEF Association: Experimental IR Meets Multilinguality, Multimodality, and Interaction. pp. 372–387. Lecture Notes in Computer Science, Springer, Avignon, France (2018)

[16] Patwari, A., Goldwasser, D., Bagchi, S.: TATHYA: a multi-classifier system for detecting check-worthy statements in political debates. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 2259–2262. CIKM '17, Singapore (2017)

[17] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv pre-print 1910.10683 (2019)

[18] Shaar, S., Nikolov, A., Babulkov, N., Alam, F., Barrón-Cedeño, A., Elsayed, T., Hasanain, M., Suwaileh, R., Haouari, F., Da San Martino, G., Nakov, P.: Overview of CheckThat! English: Automatic identification and verification of claims in social media. In: Working Notes of CLEF 2020—Conference and Labs of the Evaluation Forum. CLEF '2020, Thessaloniki, Greece (2020)

[19] Vasileva, S., Atanasova, P., Màrquez, L., Barrón-Cedeño, A., Nakov, P.: It takes nine to smell a rat: Neural multi-task learning for check-worthiness prediction. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing. pp. 1229–1239. RANLP '19, Varna, Bulgaria (2019)

[20] Wang, B., Kuo, C.C.J.: SBERT-WK: A sentence embedding method by dissecting BERT-based word models. arXiv pre-print 2002.06652 (2020)