

Check_square at CheckThat! 2020: Claim Detection in Social Media via Fusion of Transformer and Syntactic Features

Gullal S. Cheema¹, Sherzod Hakimov¹, and Ralph Ewerth^{1,2}

¹ TIB Leibniz Information Centre for Science and Technology, Hannover, Germany

² L3S Research Center, Leibniz University Hannover, Germany
{gullal.cheema, sherzod.hakimov, ralph.ewerth}@tib.eu

Abstract. In this digital age of news consumption, a news reader has the ability to react, express and share opinions with others in a highly interactive and fast manner. As a consequence, fake news has made its way into our daily life because of very limited capacity to verify news on the Internet by large companies as well as individuals. In this paper, we focus on solving two problems which are part of the fact-checking ecosystem that can help to automate fact-checking of claims in an ever increasing stream of content on social media. For the first problem, claim check-worthiness prediction, we explore the fusion of syntactic features and deep transformer Bidirectional Encoder Representations from Transformers (BERT) embeddings, to classify check-worthiness of a tweet, i.e. whether it includes a claim or not. We conduct a detailed feature analysis and present our best performing models for English and Arabic tweets. For the second problem, claim retrieval, we explore the pre-trained embeddings from a Siamese network transformer model (sentence-transformers) specifically trained for semantic textual similarity, and perform KD-search to retrieve verified claims with respect to a query tweet.

Keywords: Check-Worthiness, Fact-Checking, Social Media, Twitter, COVID-19, SVM, BERT, Retrieval, Text Classification

1 Introduction

Social media is increasingly becoming the main source of news for so many people. With around 2.5 billion Internet users, 12% receive breaking news from Twitter instead of traditional media according to a 2018 survey report [44]. Fake news in general can be defined [52] as fabrication and manipulation of information and facts with the main intention of deceiving the reader. As a result, fake news can have several undesired and negative consequences. For example,

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CLEF 2020, 22-25 September 2020, Thessaloniki, Greece.

recent news around COVID-19 pandemic with non-verified claims, that masks lead to rise in carbon dioxide levels caused an online movement to not wear masks. With ease of access and sharing news on Twitter, any news spreads much faster from the moment an event occurs in any part of the world. Although, the survey report [44] found that almost 60% of users expect news on social media to be inaccurate, it still leaves millions of people who will spread fake news expecting it to be true.

Considering the vast amount of news that spreads everyday, there has been a rise in independent fact-checking projects like *Snopes*, *Alt News*, *Our.News*, who investigate the news that spread online and publish the results for public use. Most of these independent projects rely on manual efforts that are time consuming which makes it harder to keep up with rate of news production. Therefore, it has become very important to develop tools that can process news at a rapid rate and provide news consumers with some kind of an authenticity measure that reflects the correctness of claims in the news.

In this paper, we focus on two sub-problems in CheckThat! 2020 [43]³ that are a part of larger fact-checking ecosystem. In the first task, we focus on learning a model that can recognize check-worthy claims on Twitter. We present a solution that works for both English [43] and Arabic [19] tweets. Some examples of tweets with claims are classified whether it is a check-worthy or not, shown in Table 1. One can see that the claims which are not check-worthy look like personal opinions and do not pose any threat to a larger audience. We explore the fusion of syntactic features and deep transformer Bidirectional Encoder Representations from Transformers (BERT) embeddings, to classify check-worthiness of a tweet. We use Part-of-speech (POS) tags, named entities, and dependency relations as syntactic features and a combination of hidden layers in BERT to compute tweet embedding. Before learning the model with a Support Vector Machine (SVM) [51], we use Principal Component Analysis (PCA) [57] for dimensionality reduction. In the second task, we focus on learning a model that can accurately retrieve verified claims w.r.t a query claim, where query claim is a tweet and verified claims are snippets from actual documents. The verified claim is true and thus acts as the evidence or support for the query tweet. Some example pairs of tweets and claims can be seen in Table 2, which shows that the pairs share lots of contextual information which makes this task a semantic textual similarity problem. For this reason, we explore the pre-trained embeddings from a Siamese network transformer model (sentence-transformers) specifically trained for semantic textual similarity and perform KD-search to retrieve claims. We share the source code for both tasks publicly with the community.⁴

The remainder of the paper is organized as follows. Section 2 briefly discusses about previous works on fake news detection and CheckThat! tasks in particular. Section 3 presents details of our approach for Task-1 and Task-2. Section 4 describes the experimental details and results. Section 5 summarizes our conclusion and future research directions.

³ <https://sites.google.com/view/clef2020-checkthat/>

⁴ https://github.com/cleopatra-itn/claim_detection

Table 1: Sample tweets for Task-1 Check-Worthiness Prediction

Tweet	Claim	Check-Worthy
Dear @VP Pence: What are you hiding from the American people? Why won't you let the people see and hear what experts are saying about the #CoronaOutbreak?	0	0
Greeting my good friends from the #US the #Taiwan way. Remember: to better prevent the spread of #COVID19, say no to a handshake & yes to this friendly gesture. Check it out:	0	0
Corona got these flights cheap as hell I gotta job interview in Greece Monday	1	0
My mum has a PhD on Corona Virus from WhatsApp University	1	0
This is why the beaches haven't closed in Florida, and why they've had minimal COVID-19 prevention. Absolute dysfunction. <link>	1	1
COVID-19 cases in the Philippines jumped from 24 to 35 in less than 12 hours. This is seriously ALARMING. Stay safe everyone! <link>	1	1

Table 2: Sample pairs of tweets and verified claims for Task-2 Claim Retrieval

Tweet	A number of fraudulent text messages informing individuals they have been selected for a military draft have circulated throughout the country this week.
Verified Claim	The U.S. Army is sending text messages informing people they've been selected for the military draft.
Tweet	El Paso was NEVER one of the MOST dangerous cities in the US. We've had a fence for 10 years and it has impacted illegal immigration and curbed criminal activity. It is NOT the sole deterrent. Law enforcement in our community continues to keep us safe #SOTU
Verified Claim	El Paso was one of the U.S. most dangerous cities before a border fence was built there.
Tweet	Hey @Always since today is #TransVisibilityDay it's probably important to point out the fact that this new packaging isn't trans* friendly. Just a reminder that Menstruation does not equal Female. Maybe rethink this new look. <link>
Verified Claim	"In 2019, trans activists or ""the transgender lobby"" forced Procter & Gamble to remove the Venus symbol from menstruation products."

2 Related Work

Fake news has been studied from different perspectives in the last five years, like factuality or credibility detection [38,15,42,21,20], rumour detection [61,60,47,59], propagation in networks [28,32,45,35], use of multiple modalities [27,54,48] and also as an ecosystem of smaller sub-problems like in CheckThat! [33,11,2]. For social media in particular, Shu *et al.* [46] studied and provided a comprehensive review of fake news detection with characterizations from psychology and social science, and existing computational algorithms from data mining perspective. The fact that Twitter has become a source of news for so many people, researchers have extensively used the platform to formulate problems, extract data and test their algorithms. For instance, Zubiaga *et al.* [61] extracted tweets around breaking news and used Conditional Random Fields to exploit context during the sequential learning process for rumour detection. Buntain *et al.* [6] studied three large Twitter datasets and developed models to predict accuracy assessments of fake news by crowd-sourced workers and journalists. While many approaches rely on tweet content for detecting fake news, there has been a rise in methods that exploit user characteristics and metadata to model the problem as fake news propagation. For example, Liu *et al.* [28] modeled the propagation path of each news story as a multivariate time series over users who engaged in spreading the news via tweets. They further classified the fake news using Gated Recurrent Unit (GRU) and Convolutional Neural Networks (CNN) to capture the global and local variations of user characteristics respectively. Monti *et al.* [32] went a step further and used a hybrid feature set including user characteristics, social network structure and tweet content. They modeled the problem as binary prediction using a Graph CNN resulting in a highly accurate fake news detector.

Besides fake news detection, a sub task to predict check-worthiness of claims has also been explored recently mostly in political context. For example, Hassan *et al.* [21,22] proposed a system that predicts the check-worthiness of a statement made by presidential candidates using SVM [51] classifier and combination of lexical and syntactic features. They also compared their results with fact-checking organizations like CNN⁵ and PolitiFact⁶. Later, in CheckThat! 2018 [33], several methods were proposed to improve the check-worthiness of claims in political debates. Best methods used a combination of lexical and syntactic features like Bag of Words (BoW), Parts-of-Speech (POS) tags, named Entities, sentiment, topic modeling, dependency parse trees and word embeddings [31]. Various classifiers were built using either Recurrent Neural Networks (RNN) [16,62], gradient boosting [58], k-nearest neighbor [14] or SVM [62]. In 2019 edition of CheckThat! [11], in addition to using lexical and syntactic features [13], top approaches relied on learning richer content embeddings and utilized external data for better performance. For example, Hansen *et al.* [17] used word embeddings and syntactic dependency features as input to an LSTM network,

⁵ <http://www.cnn.com>

⁶ <https://www.politifact.com/>

enriched the dataset with additional samples from Claimbuster system [23] and trained the network with a contrastive ranking loss. Favano *et. al.* [12] trained a neural network with Standard Universal Sentence Encoder (SUSE) [8] embeddings of the current sentence and previous two sentences as context. Another approach by Su *et. al.* [50] used co-reference resolution to replace pronouns with named entities to get a feature representation with bag of words, named entity similarity and relatedness. Other than political debates, Jaradat *et. al.* [25] proposed an online multilingual check-worthiness system that works for different sources (debates, news articles, interviews) in English and Arabic . They use actual annotated data from reputable fact-checking organizations and use best performing feature representations from previous approaches. For tweets in particular, Majithia *et. al.* [29] proposed a system to monitor, search and analyze factual claims in political tweets with Claimbuster [23] at the backend for check-worthiness. Lastly, Dogan *et. al.* [10] also conducted a detailed study on detecting check-worthy tweets in U.S. politics and proposed a real-time system to filter them.

3 Proposed Approach

3.1 Task-1: Tweet Check-Worthiness Prediction

Check-Worthiness prediction is the task of predicting whether a tweet includes a claim that is of interest to a large audience. Our approach is motivated by the successful use of lexical, syntactic and contextual features in the previous editions of CheckThat! check-worthiness task for political debates. Given that this task contains less amount of training data, we approached this problem with the idea of creating a rich feature representation, reducing the dimensions of large feature set with PCA [57] and then learning the model with a SVM. In doing so, our goal is also to understand which features are the most important for check-worthiness prediction from tweet content. As context is very important for downstream NLP tasks, we experiment with word embeddings (word2vec [31], GloVe [37]) and BERT [9] embeddings to create a sentence representation of each tweet. Our pre-processing and feature extraction is agnostic to the topic of the tweet so that it can be applied to any domain. Next, we provide details about all the features used, their extraction and the encoding process. Our overall approach can be seen in Figure 2.

Pre-processing We use two publicly available pre-processing tools for English and Arabic tweets. We use Baziotis *et. al.*'s [3] tool for English to apply the following normalization steps: tokenization, lower-casing, removal of punctuation, spell correction, normalize *hashtags, all-caps, censored, elongated and repeated* words, and terms like *URL, email, phone, user mentions*. We use Stanford Stanza [39] toolkit to pre-process Arabic tweets by applying the following normalization steps: tokenization, multi-word token expansion and lemmatization.

In the case of extracting word embeddings from a transformer network, we use the raw text as the networks have their own tokenization process.

Syntactic Features We use the following syntactic features for English and Arabic tasks: Parts-of-Speech (POS) tags, named entities (NE) and dependency parse tree relations. We use the pre-processed text and run off-the-shelf tools to extract syntactic information of tweets and then convert each group of information to feature sets. For English we used spaCy[24] and Stanford Stanza [39] for Arabic tweets to extract the following syntactic features. In all the features, we experiment with keeping and removing stop-words to evaluate their affect.

Part-of-Speech: For both English and Arabic, we extract 16 POS tags in total and through our empirical evaluation we find that the following eight tags to be the most useful when used as features: NOUN, VERB, PROPN, ADJ, ADV, NUM, ADP, PRON. For Arabic, the additional four tags are useful features: DET, INTJ, AUX, PART. We used the chosen set of POS tags for respective language to encode the syntactic information of tweets.

Named Entities: We identified the following named entity types to be the most important features through our evaluation: (GPE, PERSON, ORG, NORP, LOC, DATE, CARDINAL, TIME, ORDINAL, FAC, MONEY) for English and (LOC, PER, ORG, MISC) for Arabic. We also found that while developing feature combinations named entities do not add much value to overall accuracy, and hence our primary and contrastive submissions do not include them.

Syntactic Dependencies: these features are constructed using dependency relation between tokens in a given tweet. We use the dependency relation between two nodes in the parsed tree if the the child and parent nodes' POS tags are one of the following ADJ, ADV, NOUN, PROPN, VERB or NUM. All dependency relations that match the defined constraint are converted into the triplet relation such as (*child node-POS, dependency-relation, parent-POS*) and pairs such as (*child node-POS, dependency-relation*) where the relation is not part of a feature representation. This process is shown in Figure 1. We found that the features based on pairs of child and parent node perform better than the triplet feature. The dimension of the feature vector for English and Arabic is 133 and 134 respectively.

For encoding a feature, we get a histogram vector which contains the number of type of tag, named entity or syntactic relation pair. The process of feature encoding is shown in Figure 1. Finally, we normalize each type of feature with maximum value in the vector.

Average Word Embeddings One simple way to get a contextual representation of a sentence is to average the word embeddings of each token in a given sentence. For this purpose, we experiment with three types of word embeddings pre-trained on three different sources for English: GloVe embeddings [37] trained on Twitter and Wikipedia, word2vec embeddings [31] trained on Google News, and FastText [30] embeddings trained on multiple sources. In addition, we also experiment with removing stop-words from the average word representation, as stop-words can dominate in the average and result in less meaningful sentence representation. For Arabic, we use word2vec embeddings that are trained on Arabic tweets and Arabic Wikipedia [49].

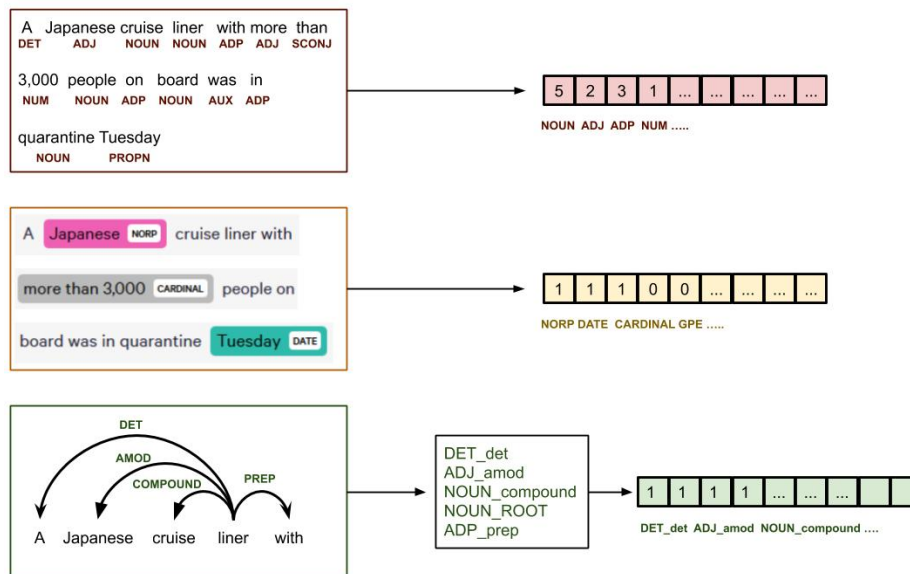


Fig. 1: Syntactic feature extraction and encoding process. Feature vectors are based on the number of times it is seen in the given sentence.

Transformer Features Another way to extract contextual features is to use BERT [9] embeddings that are trained using the context of the word in a sentence. BERT is usually trained on a very large text corpus which makes them very useful for off-the-shelf feature extraction and fine-tuning for downstream tasks in NLP. To get one embedding per tweet, we follow the observations made in [9] that, different layers of BERT capture different kinds of information, so an appropriate pooling strategy should be applied depending on the task. The paper also suggests that the last four hidden layers of the network are good for transfer learning tasks and thus we experiment with 4 different combinations, i.e. concatenate last 4 hidden layers, average of last 4 hidden layers, last hidden layer and 2nd last hidden layer. We normalize the final embedding so that l_2 norm of the vector is 1. We also experimented with BERT’s pooled sentence embedding that is encoded in the *CLS* (class) tag, which performed significantly poorer than the pooling strategies we employed. For Arabic, we only experimented with a sentence-transformer [40] that is trained on multilingual training corpus and outputs a sentence embedding for each tweet/sentence.

Sentence Representation: To get the overall representation of the tweet, we concatenate all the syntactic features together with either average word embedding or BERT-based transformer features and then apply PCA for dimensionality reduction. SVM classifier is trained on the feature vectors of tweets to output a binary decision (check worthy or not). The overall process is shown in Figure 2.

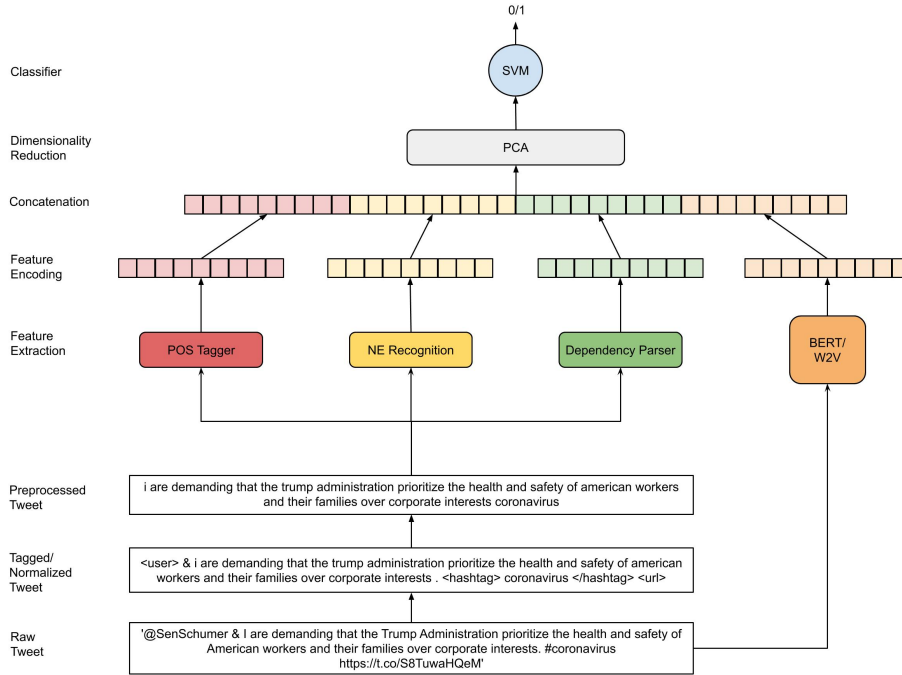


Fig. 2: Proposed Approach for Check-Worthiness Prediction

3.2 Task-2: Claim Retrieval

Claim Retrieval is the task of retrieving the most similar already verified claim to the query claim. For this task, it is important that the feature representation captures the meaning and context of words and phrases so that query matches the correct verified claim. Therefore, we relied on a triplet-network setting, where the network could be trained with triplets consisting of an anchor sample a , positive sample p and a negative sample n . We use triplet loss to fine-tune a pre-trained sentence embedding network, such that the distance between a and p is smaller than the distance between a and n using the following loss function.

$$Loss = \sum_{n=1}^N [||S_i^a - S_i^p||_2^2 - ||S_i^a - S_i^n||_2^2 + m]_+ \quad (1)$$

where S_i^a , S_i^p and S_i^n are triplet sentence embeddings and m is the margin (set to 1), N is the number of samples in the batch.

As each verified claim is a tuple consisting of text and title, we create two triplets for every true tweet-claim pair, i.e., (anchor tweet, true claim text, negative claim text) and (anchor tweet, true claim title, negative claim title). This

increases the number of positive samples for training as there are only 800 samples and one true claim for every tweet. To get negative claims, we select 3 claims with highest cosine similarity that are not the true claims for the anchor tweet using the pre-trained sentence-transformer embeddings. For pre-processing, we use Baziotis *et. al.*'s [3] tool for processing the tweets to remove *URL, email, phone, user mentions*, as the claim text or title do not contain any such information.

As retrieval is a search task, we used KD-Tree search to find the most similar already verified claim that has the minimum Euclidean distance to the query claim. The sentence embeddings extracted from the network are used to build a KD-Tree and for each query claim, top 1000 verified claims are extracted from the tree for evaluation. For building the KD-Tree, we average the sentence embeddings of claim text and claim title, as it performs better than just using either claim or title. In our ablation study, we directly compute the cosine similarity between each query tweet and all the verified claims, and pick the top 1000 (highest cosine similarity) verified claims for evaluation. We conduct the second evaluation because building a KD-Tree can affect the retrieval accuracy.

Sentence Transformers for Textual Similarity As a backbone network to extract sentence embeddings and fine-tuning with triplet loss, we use the recently proposed Sentence-BERT [9] that learns the embeddings in a Siamese (pairs) and triplet network settings. We experiment with the pre-trained Siamese Network models trained on SNLI (Stanford Natural Language Inference) [5] and STSb (Semantic Textual Similarity benchmark) [7] datasets that have been shown to perform very well for semantic textual similarity.

4 Experiments and Results

4.1 Task-1: Tweet Check-Worthiness Prediction

Dataset and Training Details English dataset consists of training, development (dev) and test splits with 672, 150 and 140 tweets respectively on the topic of COVID-19. We perform grid search using development set to find the best parameters. Arabic dataset consists of training and test splits with 1500 tweets on 3 topics and 6000 tweets on 12 topics respectively with 500 tweets on each topic. For validation purpose, we keep 10% (150 samples) from the training data as development set. The official ranking of submitted system for this task is based on Mean Average Precision (MAP) and Precision@30 (P@30) for English and Arabic datasets, respectively.

To train the SVM models for both English and Arabic, we perform grid search over PCA energy (%) conservation, regularization parameter C and RBF kernel's γ . Parameters range for PCA varies from 100% (original features) to 95% with decrements of 1, and both C and γ vary between -3 to 3 on a log-scale with 30 steps. For faster training on a large grid search, we use ThunderSVM [55] which takes advantage of a GPU or a multi-core system to speed up SVM training.

Results Our submissions used the best models that we obtained from the grid search and are briefly discussed below.

English: We made 3 submissions in total. Our primary (Run-1) and 2nd contrastive (Run-3) submission uses sentence embeddings computed from BERT-large word embeddings as discussed in the proposed work section. In addition, both submissions use POS tag and dependency relation features. Interestingly, we found that the best performing sentence embeddings did not include stop-words. The primary submission (Run-1) uses an ensemble of predictions from three models trained on concatenated last 4 hidden layers, average of last 4 hidden layers and 2nd last hidden layer. The 2nd contrastive submission (Run-3) uses predictions from the model trained on the best performing sentence embedding computed from concatenating last 4 hidden layers. Our 1st contrastive submission (Run-2) uses an ensemble of predictions from three models trained with GloVe[37] on Twitter with 25, 50 and 100-dimensional embeddings but with the same POS tag and dependency relation features. We use majority voting to get the final prediction and mean of decision values to get the final decision value. We found that removing the stop-words to compute average of word embeddings actually degraded the performance and hence included them in the average.

We also add some additional results to see the effect of stop-words, POS tags, named entities, dependency relations and ensemble predictions in Table 3. The effect of stop-words can be clearly seen in alternative runs of Run-1 and Run-3, where the MAP clearly drops by 1-2 points. Similarly, the negative effect of removing POS tag and dependency relation features can be seen in rest of the alternative runs. Lastly, adding named entity features to the original submissions also decreases the precision by 1-2 points. This might be because the tweets have very few named entities and are not useful to distinguish between check-worthy and not check-worthy claims. For comparison with other teams in the challenge, we show top 3 results at the bottom of the table for reference. Team Accenture [56] fine-tuned a RoBERTa model with an extra mean pooling and a dropout layer to prevent overfitting. Team Alex [34] experimented with different tweet pre-processing techniques and various transformer models together with logistic regression and SVM. Their main submission used logistic regression trained on 5-fold predictions from RoBERTa concatenated with tweet metadata. Team QMUL-SDS [1] fine-tuned a BERT model pre-trained specifically on COVID twitter data.

Arabic There are a total of four submissions that we made in this task. Our best performing submission (Run-1) uses 100-dimensional word2vec Arabic embeddings trained on a Twitter corpus [49] in combination with POS tag features. Our second and third submissions are redundant in terms of feature use, so we only mention the second one (Run-2) here. In addition features used in first submission, it uses dependency relation features and 300-dimensional Twitter embeddings instead of 100-dimensional. Our last submission (Run-3) uses only pre-trained multilingual sentence-transformer⁷ [41] that is trained on 10 languages including Arabic. In the first three submissions, we removed the stop-

⁷ <https://github.com/UKPLab/sentence-transformers>

Table 3: Task-1 Check-Worthiness English Results, MAP (Mean Average Precision), DRel (dependency relations), NE (named entities), *Primary Submission

Run	Stopwords	Ensemble	POS	DRel	NE	Embedding	MAP
Run-1*		✓	✓	✓		BERT	0.7217
Run-2	✓	✓	✓	✓		GloVe	0.6249
Run-3			✓	✓		BERT	0.7139
Run-1-1	✓	✓	✓	✓		BERT	0.7102
Run-1-2		✓	✓			BERT	0.6965
Run-1-3		✓				BERT	0.7094
Run-1-4		✓	✓	✓	✓	BERT	0.7100
Run-3-1	✓		✓	✓		BERT	0.6889
Run-3-2			✓			BERT	0.7074
Run-3-3						BERT	0.6981
Run-3-4			✓	✓	✓	BERT	0.6940
[56]	-	-	-	-	-	-	0.8064
[34]	-	-	-	-	-	-	0.8034
[1]	-	-	-	-	-	-	0.7141

words from all the features as keeping them resulted in a poorer performance. *Precision@K* and *Average Precision (AP)* results on the test set are shown in the same order in Table 4. Official metric for ranking is P@30. For comparison with other teams in the challenge, we show top 3 results at the bottom of the table for reference. Team Accenture [56] experimented with and fine-tuned three different pre-trained Arabic BERT models and used external data to increase the positive instances. Team TOBB-ETU [26] used logistic regression and experimented with Arabic BERT and word embeddings together to classify tweets. Team UB_ET [18] used a multilingual BERT for ranking tweets by check-worthiness.

4.2 Task-2: Claim Retrieval

Dataset and Training Details The dataset in this task has 1,003 tweets for training and 200 tweets for testing. These tweets are to be matched against a set 10,373 verified claims. From the training set, 197 tweets are kept for validation. To fine-tune the sentence-transformer network with the triplet loss, we use a batch size of eight and train the network for two epochs. The official ranking of this is based on Mean Average Precision@5 (MAP@5). All tweets and verified claims are in English.

Table 4: Task-1 Check-Worthiness Arabic Results, P@K (Precision@K) and AP (Average Precision), *Primary Submission

Run ID	P@5	P@10	P@15	P@20	P@25	P@30	AP
Run-1*	0.6000	0.6083	0.5944	0.6000	0.5900	0.5778	0.4949
Run-2	0.5500	0.5667	0.5611	0.5417	0.5433	0.5361	0.4649
Run-3	0.4000	0.3917	0.4167	0.4292	0.4433	0.4472	0.4279
[56]	0.7333	0.7167	0.7167	0.6875	0.6933	0.7000	0.6232
[26]	0.7000	0.7000	0.7000	0.6625	0.6500	0.6444	0.5816
[18]	0.6833	0.6417	0.6667	0.6333	0.6367	0.6417	0.5511

Our primary (Run-1) and 2nd contrastive (Run-3) submission uses BERT-base and BERT-large pre-trained on SNLI dataset with sentence embedding pooled from the *CLS* and *MAX* tokens respectively. We fine-tune these two networks with the triplet loss. On the contrary, our 1st contrastive submission (Run-2) uses multilingual DistilBERT model [41] trained on 10 languages including English. This model is directly used to test the pre-trained embeddings.

Results Interestingly, pre-trained embeddings extracted from multilingual DistilBERT without any fine-tuning turn out to be better for semantic similarity than fine-tuned monolingual BERT models. Having said that, the fine-tuned monolingual BERT models do perform better than extracted pre-trained embeddings and the difference can be seen in *Run-1-2* and *Run-3-2* in Table 5. We also try to fine-tune the multilingual model which drastically decreases the retrieval performance. The decrease can be attributed to the pre-training process [41] in which the model was trained in a teacher-student knowledge distillation learning framework and on multiple languages. As stated in the proposed work section, we conduct a second evaluation to retrieve the claims with highest similarity without KD-Search and the results are significantly better as shown in Table 5. For comparison with other teams in the challenge, we have shown top 3 primary submissions at the bottom of the table for reference. Team Buster.AI [4] investigated sentence similarity using transformer models, and experimented with multimodality and data augmentation. Team UNIPI-NLE [36] fine-tuned a sentence-BERT in two steps, first to predict the cosine similarity of positive and negative pairs, followed by a binary classification of whether a tweet-claim pair is a correct match or not. Team UB_ET [53] experimented with three different models to rank the verified claims and their main submission used a DPH Divergence from Randomness (DFR) term weighting model.

Table 5: Task-2 Claim Retrieval Results, MAP (Mean Average Precision), *Primary Submission

Run	Fine-tuned	KD-Search	MAP@1	MAP@3	MAP@5	MAP@10
Run-1*	✓	✓	0.6520	0.6900	0.6950	0.7000
Run-2		✓	0.8280	0.8680	0.8730	0.8740
Run-3	✓	✓	0.7180	0.7460	0.7540	0.7600
Run-1-1	✓		0.703	0.743	0.756	0.760
Run-1-2			0.527	0.584	0.589	0.594
Run-2-1			0.858	0.892	0.894	0.896
Run-3-1	✓		0.718	0.764	0.770	0.772
Run-3-2			0.532	0.569	0.576	0.585
[4]	-	-	0.8970	0.9260	0.9290	0.9290
[36]	-	-	0.8770	0.9070	0.9120	0.9130
[53]	-	-	0.8180	0.8620	0.8640	0.8660

5 Conclusion and Future Work

In this paper, we have presented our solutions for two tasks in CLEF Check-That! 2020. In the first task, we used syntactic, contextual features and SVM for predicting the check-worthiness of tweets in Arabic and English. For syntactic features, we evaluated Parts-of-Speech tags, named entities and syntactic dependency relations, and used the best feature sets for both languages. In the case of contextual features, we evaluated different word embeddings, BERT models and sentence-transformers to capture the semantics of each tweet or sentence. For future work, we would like to evaluate the possibility of using relevant meta-data and other modalities like images and videos present in tweets for claim’s check-worthiness. In the second task, we evaluated monolingual and multilingual sentence-transformers to retrieve verified claims for the query tweet. We found that off-the-shelf multilingual sentence-transformer is very well suited for semantic textual similarity task than other monolingual BERT models.

Acknowledgements

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no 812997.

References

1. Alkhalifa, R., Yoong, T., Kochkina, E., Zubiaga, A., Liakata, M.: QMUL-SDS at CheckThat! 2020: Determining COVID-19 tweet check-worthiness using an enhanced CT-BERT with numeric expressions
2. 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., Sheikh Ali, Z.: Overview of CheckThat! 2020: Automatic identification and verification of claims in social media
3. Baziotis, C., Pelekis, N., Doukeridis, C.: Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 747–754. Association for Computational Linguistics, Vancouver, Canada (August 2017)
4. Bouziane, M., Perrin, H., Cluzeau, A., Mardas, J., Sadeq, A.: Buster.AI at Check-That! 2020: Insights and recommendations to improve fact-checking.
5. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326 (2015)
6. Buntain, C., Golbeck, J.: Automatically identifying fake news in popular twitter threads. In: 2017 IEEE International Conference on Smart Cloud (SmartCloud). pp. 208–215. IEEE (2017)
7. Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L.: Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. arXiv preprint arXiv:1708.00055 (2017)
8. Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Dogan, F., et al.: Detecting Real-time Check-worthy Factual Claims in Tweets Related to US Politics. Ph.D. thesis (2015)
11. Elsayed, T., Nakov, P., Barrón-Cedeno, A., Hasanain, M., Suwaileh, R., Da San Martino, G., Atanasova, P.: Overview of the clef-2019 checkthat! lab: automatic identification and verification of claims. In: International Conference of the Cross-Language Evaluation Forum for European Languages. pp. 301–321. Springer (2019)
12. Favano, L., Carman, M.J., Lanzi, P.L.: Theearthisflat’s submission to clef’19checkthat! challenge. In: CLEF (Working Notes) (2019)
13. Gasior, J., Przybyla, P.: The ipipan team participation in the check-worthiness task of the clef2019 checkthat! lab (2019)
14. Ghanem, B., Gómez, M.M.y., Rangel, F., Rosso, P.: Upv-inaoe-autoritas-check that: Preliminary approach for checking worthiness of claims. CLEF (2018)
15. Giachanou, A., Rosso, P., Crestani, F.: Leveraging emotional signals for credibility detection. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 877–880 (2019)
16. Hansen, C., Hansen, C., Simonsen, J.G., Lioma, C.: The copenhagen team participation in the check-worthiness task of the competition of automatic identification and verification of claims in political debates of the clef-2018 checkthat! lab. In: CLEF (Working Notes) (2018)

17. Hansen, C., Hansen, C., Simonsen, J.G., Lioma, C.: Neural weakly supervised fact check-worthiness detection with contrastive sampling-based ranking loss. In: CLEF (Working Notes) (2019)
18. Hasanain, M., Elsayed, T.: bigIR at CheckThat! 2020: Multilingual BERT for ranking Arabic tweets by check-worthiness
19. Hasanain, M., Haouari, F., Suwaileh, R., Ali, Z., Hamdan, B., Elsayed, T., Barrón-Cedeño, A., Da San Martino, G., Nakov, P.: Overview of CheckThat! 2020 Arabic: Automatic identification and verification of claims in social media
20. Hassan, N., Arslan, F., Li, C., Tremayne, M.: Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1803–1812 (2017)
21. 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. pp. 1835–1838 (2015)
22. Hassan, N., Tremayne, M., Arslan, F., Li, C.: Comparing automated factual claim detection against judgments of journalism organizations. In: Computation+ Journalism Symposium. pp. 1–5 (2016)
23. Hassan, N., Zhang, G., Arslan, F., Caraballo, J., Jimenez, D., Gawsane, S., Hasan, S., Joseph, M., Kulkarni, A., Nayak, A.K., et al.: Claimbuster: the first-ever end-to-end fact-checking system. Proceedings of the VLDB Endowment **10**(12), 1945–1948 (2017)
24. Honnibal, M., Montani, I.: spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. To appear (2017)
25. Jaradat, I., Gencheva, P., Barrón-Cedeño, A., Márquez, L., Nakov, P.: Claimrank: Detecting check-worthy claims in arabic and english. arXiv preprint arXiv:1804.07587 (2018)
26. Kartal, Y.S., Kutlu, M.: TOBB ETU at CheckThat! 2020: Prioritizing English and Arabic claims based on check-worthiness
27. Khattar, D., Goud, J.S., Gupta, M., Varma, V.: Mvae: Multimodal variational autoencoder for fake news detection. In: The World Wide Web Conference. pp. 2915–2921 (2019)
28. Liu, Y., Wu, Y.F.B.: Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
29. Majithia, S., Arslan, F., Lubal, S., Jimenez, D., Arora, P., Caraballo, J., Li, C.: Claimportal: Integrated monitoring, searching, checking, and analytics of factual claims on twitter. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 153–158 (2019)
30. Mikolov, T., Grave, E., Bojanowski, P., Puhersch, C., Joulin, A.: Advances in pre-training distributed word representations. arXiv preprint arXiv:1712.09405 (2017)
31. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
32. Monti, F., Frasca, F., Eynard, D., Mannion, D., Bronstein, M.M.: Fake news detection on social media using geometric deep learning. arXiv preprint arXiv:1902.06673 (2019)
33. Nakov, P., Barrón-Cedeno, A., Elsayed, T., Suwaileh, R., Márquez, L., Zaghouni, 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: International Conference of the Cross-Language Evaluation Forum for European Languages. pp. 372–387. Springer (2018)
34. Nikolov, A., Da San Martino, G., Koychev, I., Nakov, P.: Team_Alex at CheckThat! 2020: Identifying check-worthy tweets with transformer models
 35. Papanastasiou, Y.: Fake news propagation and detection: A sequential model. *Management Science* **66**(5), 1826–1846 (2020)
 36. Passaro, L., Bondielli, A., Lenci, A., Marcelloni, F.: UNIPI-NLE at CheckThat! 2020: Approaching fact checking from a sentence similarity perspective through the lens of transformers
 37. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
 38. Popat, K., Mukherjee, S., Strötgen, J., Weikum, G.: Credibility assessment of textual claims on the web. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 2173–2178 (2016)
 39. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2020)
 40. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
 41. Reimers, N., Gurevych, I.: Making monolingual sentence embeddings multilingual using knowledge distillation. arXiv preprint arXiv:2004.09813 (04 2020), <http://arxiv.org/abs/2004.09813>
 42. Samadi, M., Talukdar, P., Veloso, M., Blum, M.: Claimeval: Integrated and flexible framework for claim evaluation using credibility of sources. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
 43. 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! 2020 English: Automatic identification and verification of claims in social media
 44. Shearer, E., Eva Matsa, K.: How Social Media Has Changed How We Consume News (2018), <https://www.journalism.org/2018/09/10/news-use-across-social-media-platforms-2018/>
 45. Shu, K., Bernard, H.R., Liu, H.: Studying fake news via network analysis: detection and mitigation. In: Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining, pp. 43–65. Springer (2019)
 46. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter* **19**(1), 22–36 (2017)
 47. Sicilia, R., Giudice, S.L., Pei, Y., Pechenizkiy, M., Soda, P.: Twitter rumour detection in the health domain. *Expert Systems with Applications* **110**, 33–40 (2018)
 48. Singhal, S., Shah, R.R., Chakraborty, T., Kumaraguru, P., Satoh, S.: Spofake: A multi-modal framework for fake news detection. In: 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM). pp. 39–47. IEEE (2019)
 49. Soliman, A.B., Eissa, K., El-Beltagy, S.R.: Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science* **117**, 256–265 (2017)
 50. Su, T., Macdonald, C., Ounis, I.: Entity detection for check-worthiness prediction: Glasgow terrier at clef checkthat! 2019 (2019)

51. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. *Neural processing letters* **9**(3), 293–300 (1999)
52. Tandoc Jr, E.C., Lim, Z.W., Ling, R.: Defining “fake news” a typology of scholarly definitions. *Digital journalism* **6**(2), 137–153 (2018)
53. Thuma, E., Motlogelwa, N.P., Leburu-Dingalo, T., Mudongo, M.: UB.ET at CheckThat! 2020: Exploring ad hoc retrieval approaches in verified claims retrieval
54. Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., Su, L., Gao, J.: Eann: Event adversarial neural networks for multi-modal fake news detection. In: *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*. pp. 849–857 (2018)
55. Wen, Z., Shi, J., Li, Q., He, B., Chen, J.: ThunderSVM: A fast SVM library on GPUs and CPUs. *Journal of Machine Learning Research* **19**, 797–801 (2018)
56. Williams, E., Rodrigues, P., Novak, V.: Accenture at CheckThat! 2020: If you say so: Post-hoc fact-checking of claims using transformer-based models
57. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. *Chemometrics and intelligent laboratory systems* **2**(1-3), 37–52 (1987)
58. Yasser, K., Kutlu, M., Elsayed, T.: bigir at clef 2018: Detection and verification of check-worthy political claims. In: *CLEF (Working Notes)* (2018)
59. Zhao, Z., Resnick, P., Mei, Q.: Enquiring minds: Early detection of rumors in social media from enquiry posts. In: *Proceedings of the 24th international conference on world wide web*. pp. 1395–1405 (2015)
60. Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., Procter, R.: Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)* **51**(2), 1–36 (2018)
61. Zubiaga, A., Liakata, M., Procter, R.: Exploiting context for rumour detection in social media. In: *International Conference on Social Informatics*. pp. 109–123. Springer (2017)
62. Zuo, C., Karakas, A., Banerjee, R.: A hybrid recognition system for check-worthy claims using heuristics and supervised learning. In: *CLEF (Working Notes)* (2018)