CrowdChecked: Detecting Previously Fact-Checked Claims in Social Media

Anonymous ACL submission

Abstract

While there has been substantial progress in developing systems to automate the process of fact-checking, such systems still lack credibility in the eyes of the users, and thus human fact-checkers remain the main drivers of the process. In view of that, recently, a middle-ground approach has emerged: to do automatic fact-checking by verifying whether the input claim has been previously fact-checked by professional fact-checkers, and to return back an article that explains the verdict on the claim. This is a sensible approach as people trust manual fact-checking, and as many claims are repeated multiple times online. Yet, a major issue when building such kinds of systems is the small number of known input–verified claim pairs available for training. Here, we aim to bridge this gap by making use of crowd fact-checking, i.e., mining claims in social media for which users have responded with a link to a fact-checking article. In particular, we mine a large-scale collection of 330,000 tweets paired with a corresponding fact-checking article. We further propose a new model to learn from this noisy data based on modified self-adaptive training, in a distant supervision scenario. Our experiments on a standard test set show improvements over the state of the art by two points absolute.

1 Introduction

The massive spread of disinformation online, especially in social media, was counter-acted by major efforts to limit the impact of false information not only by journalists and fact-checking organizations but also by governments, private companies, researchers, and ordinary Internet users. Such efforts include, but are not limited to building systems for automatic fact-checking (Thorne and Vlachos, 2018; Guo et al., 2021), rumor debunking (Zubiaga et al., 2016a; Derczynski et al., 2017), fake news detection (Ferreira, 2016; Pomerleau and Rao, 2017), and media profiling (Baly et al., 2020; Stefanov et al., 2020), among others.

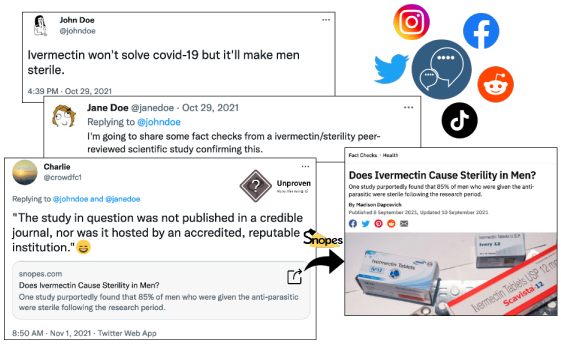

Figure 1: Crowd fact-checking thread on Twitter.

We study the problem of detecting previously fact-checked claims, which is an integral part of an end-to-end fact-checking pipeline (Hassan et al., 2017), and also an important task on its own right as people often repeat the same claim online (Barrón-Cedeno et al., 2020; Vo and Lee, 2020; Shaar et al., 2021). Unfortunately, research on this problem is limited by data scarceness, with datasets typically having about a 1,000 input–verified claim pairs (Barrón-Cedeno et al., 2020; Shaar et al., 2020, 2021), with the notable exception of Vo and Lee (2020), which contains 19K claims about images matched against 3K fact-checking articles.

We propose to bridge this gap using crowd fact-checking to create a large collection of tweet–article pairs, which we then label automatically using distant supervision. An example is shown in Figure 1, where the first two tweets discuss a controversial claim, while the third tweet offers a link to a corresponding fact-checking article.

Our contributions are as follows:

- we mine a large-scale collection of 330,000 tweets paired with fact-checking articles;
- we propose two distant supervision strategies to label the dataset;
- we propose a novel approach to learn from this data using a modified self-adaptive training;
- we demonstrate sizable improvements over the state of the art on a standard test set.

2 Related Work

Previously Fact-Checked Claims While fake news and mis/disinformation detection have been studied extensively (Zubiaga et al., 2016b; Li et al., 2016; Zubiaga et al., 2018; Martino et al., 2020; Hardalov et al., 2021; Guo et al., 2021), the problem of detecting previously fact-checked claims remains under-explored. Hassan et al. (2017) mentioned the task as a component of their end-to-end fact-checking pipeline, but did not evaluate it in isolation, neither did they study its contribution.

Recently, the task received more attention from the research community. Shaar et al. (2020) collected two datasets, from PolitiFact (political debates) and from Snopes (tweets), of claim and corresponding fact-checking articles. The CLEF *CheckThat!* lab (Barrón-Cedeno et al., 2020; Shaar et al., 2021) extended these datasets with additional data in English and Arabic. The best-performing systems (Pritzkau, 2021; Mihaylova et al., 2021; Chernyavskiy et al., 2021a) used a combination BM25 retrieval, semantic similarity using sentence embeddings (Reimers and Gurevych, 2019), and reranking. Bouziane et al. (2020) further used external data from fact-checking datasets (Wang, 2017; Thorne et al., 2018; Wadden et al., 2020).

Our work is most similar to that of Vo and Lee (2020), who mined 19K tweets and corresponding fact-checked articles. Unlike them, we focus on textual claims (they were interested in multimodal tweets with images), we collect an order of magnitude more examples, and we propose a novel approach to learn from such noisy data directly (while they manually checked each example).

Training with Noisy Data Levering large collections of unlabeled data has been at the core of large-scale language models, such as GPT (Radford et al., 2018, 2019), BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019). Recently, such language models used noisy retrieved data (Lewis et al., 2020; Guu et al., 2020) or active relabeling and data augmentation (Thakur et al., 2021). Moreover, using distantly supervised data labeling is a crucial part of the recent breakthroughs in few-shot learning (Schick and Schütze, 2021a,b).

Yet, there has been little work of using noisy data for fact-checking tasks. Vo and Lee (2019) collected tweets containing a link to a fact-checking website, based on which tried to learn a fact-checking language and to generate automatic answers. You et al. (2019) used similar data from tweets for fact-checking URL recommendations.

Unlike the above work, here we propose an automatic procedure for labeling and self-training specifically designed for the task of detecting previously fact-checked claims.

3 Our Dataset: CrowdChecked

3.1 Dataset Collection

We use Snopes as our target fact-checking website, due to its popularity among both Internet users and researchers (Popat et al., 2016; Hanselowski et al., 2019; Augenstein et al., 2019; Tchechmedjiev et al., 2019). We further use Twitter as the source for collecting user messages, which could contain claims and fact-checks of these claims.

Our data collection setup is similar to the one in (Vo and Lee, 2019). First, we form a query[1] to select tweets that contain a link to a fact-check from Snopes (*url:snopes.com/fact-check/*), which is either a reply or a quote tweet, and not a retweet.[2] We analyze in more detail the conversation structure of these fact-checked tweets in Appendix B.1.

We then collect all tweets that match our query in the interval from October 2017 till October 2021, which yielded a total of 482,736 unique hits. We further collect 148,503 reply tweets and 204,250 conversation (root) tweets.[3] Finally, we filter out malformed pairs, i.e., tweets linking to themselves, empty tweets, non-English results, tweets with no resolved URLs in the Twitter object (*'entities'*), and tweets with broken links to the fact-checking website. After cleaning the dataset, we ended up with 332,660 unique tweet–article pairs (shown in first row in Table 4), 316,564 unique tweets, and 10,340 fact-checking articles from Snopes they could point to. More detail about the fact-checking articles collection and statistics are given in Appendix B.2 and on Figure 2.

3.2 Comparison to Existing Datasets

Next, we compare our dataset to a closely related dataset from the CLEF-2021 CheckThat '21 lab Task 2A on Detecting Previously Fact-Checked Claims in Tweets (Shaar et al., 2021), to which we will refer as *CheckThat '21* in the rest of the paper.

[1] We use the Twitter API v2 with academic research access.

[2] We exclude retweets, as they do contain no comments, but rather share previous tweets.

[3] The sum of the unique replies and of the conversation tweets is not equal to the number of fact-checking tweets, as more than one tweet might reply to the same comment.

Dataset	Tweets‡	Words			Vocab
	|Unique|	Mean	50%	Max	|Unique|
CrowdChecked (Ours)	316,564	12.2	11	60	114,727
CheckThat '21	1,399	17.5	16	62	9,007

Table 1: Statistics about our dataset vs. CheckThat '21. ‡The number of unique tweets is lower compared to the total number of tweet–article pairs, as one tweet can be fact-checked by multiple articles.

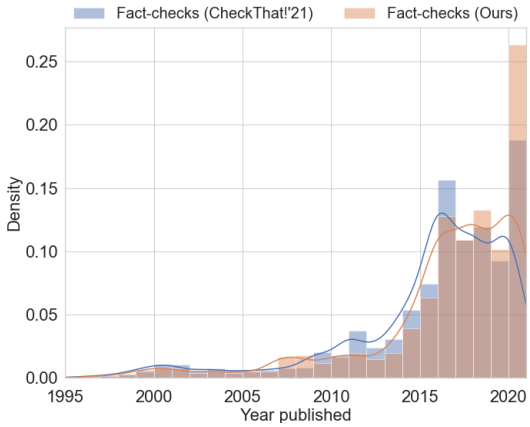

Figure 2: Histogram of the year of publication of the Snopes articles included in *CrowdChecked* (our dataset) vs. those in *CheckThat '21*.

All other datasets related to our task are either smaller (Barrón-Cedeno et al., 2020), come from a different domain (Shaar et al., 2021), are not in English (Elsayed et al., 2019), or are multi-modal (Vo and Lee, 2020).

Table 1 compares our *CrowdChecked* to *CheckThat '21* in terms of number of examples, length of the tweets, and vocabulary size. Before we calculated these statistics, we lowercased the text and we removed all URLs, Twitter handlers, English stop words, and punctuation. We can see in Table 1 that *CrowdChecked* contains two orders of magnitude more examples, slightly shorter tweets (but the maximum length stays approximately the same, which can be explained by the word limit of Twitter), and has a vocabulary size that is an order of magnitude larger. Note, however, that many examples in *CrowdChecked* are not good matches (see Section 3.1), and thus we use distant supervision to label them (see Section 3.3), with the resulting dataset sizes of matching pairs shown in Table 4.

Finally, we compare the set of Snopes fact-checking articles referenced by the crowd fact-checkers to the ones included in the CheckThat '21 competition. We can see that the tweets in CrowdChecked refer to around 3.5K less articles (namely 10,340), compared to CheckThat '21, which consists of 13,835 articles. A total of 8,898 articles are present in both datasets. Since the CheckThat '21 is collected earlier, it includes less articles from recent years compared to CrowdChecked, and peaks at 2016/2017. Nevertheless, for CheckThat '21, the number of Snopes articles included in a claim–article pair is far less compared to our dataset (even after filtering out the unrelated pairs), as it is capped at the number of tweets included in that dataset (which is 1.4K).

More detail about the process of collecting the fact-checking articles is given in Appendix B.2.

3.3 Data Labeling (Distant Supervision)

To label our examples, we experiment with two distant supervision approaches: (*i*) based on the Jaccard similarity between the tweet and its fact-checking article, and (*ii*) based on the predictions of a model trained on CheckThat '21. We describe these two approaches in more detail below.

Jaccard Similarity In this approach, we first preprocess the texts by converting them to lowercase, removing all URLs and replacing all numbers with a single zero. Then, we tokenize the texts using the NLTK's *Twitter tokenizer* (Loper and Bird, 2002), and we strip all handles and user mentions. The final preprocessing step is to filter out all stop words and punctuation (including quotes and special symbols) and to stem (Porter, 1980) all tokens.

In order to obtain a numerical score for each tweet–article pair, we calculate the *Jaccard similarity* (jac) between the normalized tweet text and each of the *title* and the *subtitle* from the Snopes article (i.e., the intersection over the union of the unique tokens). Both fields present a summary of the fact-checked claim, and thus should include more compressed information. Finally, we average these two similarity values to obtain a more robust score. Statistics are shown in Table 2.

Semi-Supervision Here, we train a Sentence-BERT (Reimers and Gurevych, 2019) model, as described in Section 4, using the manually annotated data from CheckThat '21. The model shows strong performance on the testing set of CheckThat '21 (see Table 5), and thus we expect it to have good precision at detecting matching fact-checked pairs. In particular, we calculate the *cosine similarity* be-

Range (Jaccard)	Examples (%)	Good Pairs Reply (%)	Good Pairs Conv. (%)
[0.0;0.1)	62.57	5.88	0.00
[0.1;0.2)	18.98	36.36	14.29
[0.2;0.3)	10.21	46.67	50.00
[0.3;0.4)	4.17	76.47	78.57
[0.4;0.5)	2.33	92.86	92.86
[0.5;0.6)	1.08	94.12	94.12
[0.6;0.7)	0.43	80.00	80.00
[0.7;0.8)	0.11	92.31	92.31
[0.8;0.9)	0.05	91.67	92.86
[0.9;1.0]	0.02	100.00	100.00

Table 2: Proportion of examples in different bins based on average Jaccard similarity between the tweet ↔ the title/subtitle. Manual annotations of good pairs.

Range (Cosine)	Examples (%)	Good Pairs (%)
[-0.4;0.1)	37.83	0.00
[0.1;0.2)	16.50	6.67
[0.2;0.3)	12.28	41.46
[0.3;0.4)	10.12	36.36
[0.4;0.5)	8.58	63.16
[0.5;0.6)	6.69	70.00
[0.6;0.7)	4.47	84.21
[0.7;0.8)	2.48	96.15
[0.8;0.9)	0.97	93.10
[0.9;1.0]	0.08	100.00

Table 3: Proportion of examples in different bins based on cosine similarity using sentence-BERT trained on *CheckThat '21*. Manual annotations of good pairs.

tween the embeddings of the fact-checked tweet and the fields from the Snopes article. Statistics about the scores are shown in Table 3.

3.4 Feasibility Evaluation

To evaluate the feasibility of the obtained labels, we performed manual annotation, aiming to estimate the number of *good pairs* (i.e., tweet–article pairs, where the article fact-checks the claim in the tweet). Our prior observations of the data suggested that unbiased sampling from the pool of tweets was not suitable, as it would include mostly pairs that have very few overlapping words, which is often an indicator that the texts are not related. Thus, we sample the candidates for annotation based on their Jaccard similarity, i.e., we divided the range of possible values [0;1] into 10 equally sized bins and we sampled 15 examples from each bin, resulting into 150 conversation–reply–tweet triples. Afterwards, the appropriateness of each reply-article and conversation-article pair is annotated by three annotators independently. The annotators had a *good level* of inter-annotator agreement: 0.750 for the conversations, and 0.745 for the replies in terms of Fleiss Kappa (Fleiss, 1971) (see Appendix C).

Tables 2 and 3 show the resulting estimates of *good pairs* for both Jaccard and cosine-based labeling. In the case of Jaccard, we can see that the expected number of good examples is very high (over 90%) in the range of *[0.4–1.0]*, and then it drastically decreases, going to almost zero when the similarity is less than 0.1. Similarly, for the cosine score, we can see high number of matches in the top 4 bins (*[0.6–1.0]*), albeit the number of matches remains relatively high in the following interval of *[0.2–0.6)* between 36% and 63%, and again gets close to zero for the lower-score bins. We analyze the distribution of the Jaccard scores in CheckThat '21 in more detail in Appendix B.3.

4 Method

General Scheme As a base for our models, we use Sentence-BERT (SBERT). It uses a Siamese network trained with a Transformer (Vaswani et al., 2017) encoder to obtain sentence-level embeddings. We keep the base architecture proposed by Reimers and Gurevych (2019), but we use additional features, training tricks, and losses described in the next sections. The input of the model is a pair of a tweet and fact-checking article, which we encode as follows:

- User Tweet:
  [CLS] *Tweet Text* [SEP]
- Fact-checking article:
  [CLS] *Title* [SEP] *Subtitle* [SEP] *Verified Claim* [SEP]

We train the models using the Multiple Negatives Ranking (MNR) loss (Henderson et al., 2017) (see Eq. 1), instead of the standard cross-entropy loss, as the datasets contain only positive (i.e., matching) pairs. Moreover, we propose a new variant of the MNR loss that accounts for the noise in the dataset, as described in detail in Section 4.1.

Enriched Scheme In the enriched scheme of the model, we adopt the pipeline proposed in the best-performing system from the CheckThat '21 competition (Chernyavskiy et al., 2021b). Their method consists of independent components for

assessing lexical (TF.IDF-based) and semantic (SBERT-based) similarities. The SBERT models use the same architecture and input format as described in the *'General Scheme'* above. However, Chernyavskiy et al. (2021b) use an ensemble of models, i.e., instead of calculating a single similarity between the tweet and the joint title/subtitle/verified claim, the similarities between the tweet and the claim, the joint title/claim, and the three together are obtained from three models, one using on TF.IDF and one using SBERT, for each combination. These similarities are combined via a re-ranking model (see Section 4.2. In our experiments, the TF.IDF and the model ensembles are included only in the models with re-ranking.

Shuffling and Temperature Additionally, we adopt a temperature parameter (τ) in the MNR loss. We also make it trainable in order to stabilize the training process as suggested in (Chernyavskiy et al., 2021a). This forces the loss to focus on the most complex and important examples in the batch. Moreover, this effect is amplified after each epoch by an additional data shuffling that composes batches from several groups of the most similar examples. This shuffling, in turn, increases the temperature significance. The nearest neighbors forming the groups are found using the model predictions. More detail can be found in (Chernyavskiy et al., 2021b).

4.1 Training with Noisy Data

Self-Adaptive Training To account for possible noise in the distantly supervised data, we modify the training process and apply a self-adaptive training (Huang et al., 2020). We iteratively refurbish the labels y using the predictions of the current model starting after an epoch of choice, which is a hyper-parameter:

$y^r \leftarrow \alpha \cdot y^r + (1-\alpha) \cdot \hat{y}$,

where y^r is the current refurbished label ($y_r = y$ initially), $\hat{y}$ is the model prediction, and α is a momentum hyper-parameter (we set α to 0.9).

Since the MNR loss operates with positive pairs only (it does not operate with labels), to implement this approach, we had to modify the loss function. Let $\{c_i, v_i\}_{1..m}$ be the batch of input pairs, where m is the batch size, $C, V \in \mathbb{R}^{m \times h}$ are the matrices of embeddings for the tweets and for the fact-checking articles (h is the embeddings' hidden size), and C, V are normalized to the unit hypersphere (we use cosine similarity), then:

$$\mathcal{L} = -\frac{1}{m}\sum_{i=1}^{m} {y^r}_i \Big(\frac{c_i^T v_i}{\tau} - \log \sum_{j=1}^{m} \exp(\frac{c_i^T v_j}{\tau}) \Big) \quad (1)$$

If we set $y_i^r = 1$, then Eq. 1 resembles the MNR loss definition. The parameter τ is the temperature, discussed in Section 4 *Shuffling and Temperature*.

Weighting In the self-adaptive training approach, Huang et al. (2020) introduce weights $w_i = \max_{j \in \{1,..,L\}} t_{i,j}$, where t_i is the corrected one-hot encoded target vector in a classification task with L classes. The goal is to ensure that noisy labels will have a lower influence on the training process compared to correct labels. Instead of a classification task with one-hot target vectors $t_{i,j}$, here we have real targets y_i^r. Therefore, we take these probabilities as weights: $w_i = y_i^r$. After applying both modifications with the addition of labels and weights, the impact of each training example is proportional to the square of the corrected label, i.e., in Eq. 1 y_i^r is now squared.

4.2 Re-ranking

Re-ranking has shown major improvements for detecting previously fact-checked claims (Shaar et al., 2020, 2021; Mihaylova et al., 2021; Chernyavskiy et al., 2021b), and thus we include it as part of our model. In particular, we adopt the re-ranking procedure from Chernyavskiy et al. (2021b). It uses a LambdaMART (Wu et al., 2010) model. The inputs are the reciprocal ranks (position in the ranked list of claims) and the predicted relevance scores (2 factors) based on the scores of the TF.IDF and S-BERT models (2 models), between the tweet and the claim, claim+title, and claim+title+subtitle (3 combinations), for a total of 12 features in the ensemble and 4 in the single model.

5 Experiments

In this section, we describe our experimental setup and we present our experimental results. The training procedure and the hyper-parameters are discussed in Appendix A, and the baseline models are in Appendix D.1.

5.1 Experimental Setup

Datasets Table 4 shows statistics about the data split sizes for CrowdChecked and CheckThat '21. We use these splits in our experiments, albeit sometimes mixed together.

Dataset	Data Split	Threshold	Tweet-Article Pairs
CrowdChecked (Our Dataset)	Train	-	332,660
	Train *Jaccard*	0.30	27,387
		0.40	12,555
		0.50	4,953
	Train *Cosine*	0.50	48,845
		0.60	26,588
		0.70	11,734
		0.80	3,496
CheckThat '21	Train	-	999
	Dev[4]	-	199
	Test	-	202

Table 4: Statistics about our collected datasets in terms of tweet–Aaticle pairs. Each subset is used for training.

Model	MRR	P@1	MAP@5
Baselines (CheckThat '21)			
Retrieval (Shaar et al., 2021)	76.1	70.3	74.9
SBERT (CheckThat '21)	79.96	74.59	79.20
CrowdChecked (Our Dataset)			
SBERT (jac > 0.30)	81.50	76.40	80.84
SBERT (cos > 0.50)	81.58	75.91	81.05
(Pre-train) CrowdChecked, (Fine-tune) CheckThat '21			
SBERT (jac > 0.30, Seq)	**83.76**	**78.88**	**83.11**
SBERT (cos > 0.50, Seq)	82.26	77.06	81.41
(Mix) CrowdChecked and CheckThat '21			
SBERT (jac > 0.30, Mix)	83.04	78.55	82.30
SBERT (cos > 0.50, Mix)	82.12	76.57	81.38

Table 5: Evaluation on the CheckThat '21 testing set. In parenthesis is name of the training split, i.e., *Jac*card or *Cos*ine selection strategy, *(Seq)* first training on CrowdChecked and then on CheckThat '21, *(Mix)* mixing the data from the two. The highest results are in **bold**.

The first group (CrowdChecked) is the data splits obtained from distant supervision. As the positive pairs are annotated with distant supervision and not by humans, we only include the examples as part of the training set. Each shown split is obtained using a different similarity measure (Jaccard or Cosine) or threshold (see Section 3.3). From the total number of 332,660 collected tweet–claim pairs in CrowdChecked, we end up with subsets of sizes between 3.5K and 49K examples.

The second group describes the CheckThat '21 dataset. We preserve the original training, development, and testing splits. In each of our experiments, we validate and test on the corresponding subsets from the CheckThat '21, while the training set can be a mix with CrowdChecked.

Metrics For our evaluation, we adopt the ranking measures used in the CheckThat '21 competition. In particular, we calculate the Mean Reciprocal Rank (MRR), Mean Average Precision (MAP@K) and Precision@K, for $K \in \{1, 3, 5, 10\}$. All the models are optimized for MAP@5, as was in the CLEF-2021 CheckThat! lab subtask 2A.

5.2 Experimental Results

Below, we present experiments that (*i*) aim to analyze the impact of training with the distantly supervised data from CrowdChecked, and (*i*) to further improve the state-of-the-art (SOTA) results using modeling techniques to better leverage the noisy data points (see Section 4). In all our experiments, we evaluate the model on the development and on the testing sets from CheckThat '21 (see Table 4), and we train on a mix with *CrowdChecked*. The reported results for each experiment (for each metric) are averaged over three runs using different seeds.

Data Efficiency Our goal here is to evaluate the impact of using distantly supervised data from CrowdChecked. In particular, we train an SBERT baseline, as described in Section 4, using four different training datasets: (*i*) the training data from CheckThat '21, (*ii*) training data from *CrowdChecked*, (*iii*) pre-training on data from *CrowdChecked* and then fine-tuning on CheckThat '21, (*iv*) mixing the data from both datasets.

Table 5 shows the results grouped based on training data used. In each group, we include the two best-performing models. We see that all SBERT models outperform the Retrieval baseline by 4–8 points absolute in terms of MAP@5. Interestingly, training only on distantly supervised data is enough to outperform the SBERT trained on the CheckThat '21 dataset by more than 1.5 MAP@5 points absolute. Moreover, the performance of both data labeling strategies (i.e., Jaccard and Cosine) is relatively close, suggesting comparable amount of noise in the two datasets.

Next, we train on combined data from the two datasets. Unsurprisingly, both mixing the data and training on the two datasets sequentially (first on CrowdChecked and then on CheckThat '21) yields additional improvement compared to training on a single dataset. Moreover, we observe the best

[4] Shaar et al. (2021) lists 200, but there is one duplicate row in the development set.

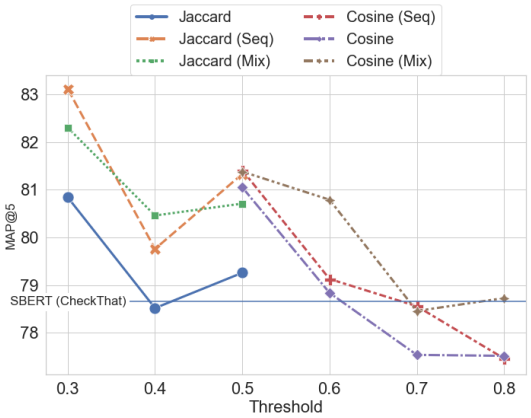

Figure 3: MAP@5 for different thresholds and distant supervision approaches. *Jaccard, Cosine* models are trained only on CrowdChecked, (*Seq*) and (*Mix*) – also on CheckThat '21.

Model	**MAP@5**	
	Dev	Test
DIPS (Mihaylova et al., 2021)	93.6	78.7
NLytics (Pritzkau, 2021)	-	79.9
Aschern (Chernyavskiy et al., 2021b)	94.2	88.2
SBERT (jac > 0.30, Mix)	90.0	82.3
+ shuffling & trainable temp.	92.4	82.6
+ self-adaptive training (Eq. 1)	92.6	83.6
+ loss weights	*92.7*	*84.3*
+ TF.IDF + Re-ranking	93.1	89.7
+ TF.IDF + Re-ranking (ens.)	**94.8**	**90.3**

Table 6: Results on CheckThat '21 (dev and test). We compare our model and its components (added sequentially) to the state of the art. The best results are in **bold**.

result when the model is first pre-trained on the *(Jaccard > 0.3)* subset of CrowdChecked, and then fine-tuned on CheckThat '21. This combination gains 2 points absolute in terms of MRR, P@1, and MAP@5, compared to *SBERT (CrowdChecked)* and 4 points compared to *SBERT (CheckThat '21)*. Nevertheless, we must note that pre-training with the *Cosine similarly (cos > 0.50)* did not yield such sizable improvements as the ones when using Jaccard. We attribute this, on one hand, to the more expected noise in the data according to our manual annotations (see Section 3.4), and on the other, to the fact that these examples are annotated by a similar model, so they are presumably easy for it.

Further, we analyze the effects of choosing different thresholds for the distant supervision approaches. Figure 3 shows the change of MAP@5 for each data labeling strategy. On the left part of the figure, in the interval [0.3–0.5], are shown the results of the Jaccard-based data labeling strategy, and on the right ([0.5–0.8]) – the Cosine strategy. Once again, the models trained on the data selected using Jaccard similarity perform similarly or better as the *SBERT (CheckThat '21)* model (blue solid line). On the other hand, the Cosine-based selection outperforms the baseline only in small thresholds $\leq$ 0.6. These observations are in favor of the hypothesis that the highly ranked pairs from the fine-tuned SBERT model are easy examples, and do not bring much signal to the model over the CheckThat '21 data, whereas the Jaccard ranked ones significantly improve the model's performance. Nonetheless, we see similar performance when training with data from the lowest two thresholds for the two similarities (without data mixing), which suggests that these subsets have similar characteristics.

Adding more distantly supervised data is beneficial for the model, regardless of the strategy. The only exception is the drop in performance when we decrease the Jaccard threshold from 0.5 to 0.4. We attribute this to the quality of the data in that bracket, as the examples with lower similarity are expected to add more noise, however the results improve drastically at the next threshold (adding x2 more examples). The latter suggests that the model was able to generalize better from the new data. There is no such drop in the Cosine strategy. We explain this with expectation that noise increases proportionally to the decrease in model confidence.

Finally, we report the performance of each model both on the development and testing sets in Appendix D.2, Tables 9 and 10.

Modeling Noisy Data We explore the effects of the proposed changes to the SBERT training approach: (*i*) shuffling and training temperature, (*ii*) data-related modification of the MNR loss for self-adaptive training with weights. We use the *(jac > 0.30, mix)* approach in our experiments, as the baseline SBERT models achieved the highest scores on the development set (Table 9). In Table 6, we ablate each of these modifications by adding them iteratively to the baseline SBERT model.

First, we can see that adding a special shuffling procedure and a trainable temperature (τ) improves the MAP@5 by 2 points on the dev set and 0.3 on the test set. Next, we see a sizable improvement of 1 point MAP@5 on the test set, when using the self-adaptive training with MNR loss. More-

over, an additional 0.7 points comes from adding weights to the loss, arriving at 84.3 MAP@5. These weights allow the model to give higher importance to the less noisy data during the training process. Here, we must note that for these two ablations the improvements on the development set are diminishing. We attribute this to its small size (199 examples) and the high values of MAP@5. Finally, note that our model, without using re-ranking, outperforms all state-of-the-art models, except Aschern, by more than 4.5 points on the testing dataset.

On the last two rows of Table 6, we present the results of our model that includes all proposed components, in combination with TF.IDF features and the LambdaMART re-ranking, described in Section 4. Here, we must note that the model is trained on a part of the CheckThat '21 training pool (80%) – the other part is used to train the re-ranking model. The full setup boosts the model's MAP@5 up to *89.7* when using a single model of the TF.IDF and SBERT (using the title/subtitle/claim as inputs, same as SBERT). With the ensemble architecture (re-ranking based on the scores of three TF.IDF and three SBERT models), we reach our best results of *90.3* on the test set (adding 1.7 MAP@5 on dev, and 0.6 on test), outperforming the previous state-of-the-art approach (Aschern Chernyavskiy et al. (2021b), 88.2) by 2 points MAP@5, and more than 11 compared to the second best model (Pritzkau (2021), 79.9). This improvement corresponds to the observed gain over the SBERT model without re-ranking. Nevertheless, the change in the strength of the factors in LambdaMART is less. The TF-IDF models still have high importance for re-ranking – a total of 41% compared to 42.8% reported in Chernyavskiy et al. (2021b). Here, we have a decrease mainly due to an increase of the importance of the reciprocal rank factor from 18.8% to 20.2% of the SBERT model that selects candidates. The strength of other factors remains almost unchanged.

6 Discussion

Our proposed distant supervision data selection strategies show promising results, achieving SOTA results on the CheckThat '21. Nonetheless, we are not able to identify all matching pairs in the list of candidates in CrowdChecked. Hereby, we try to estimate their expected number using the statistics from our manual annotations,[5] shown in Tables 2,3.

In particular, we estimate it by multiplying the fraction of good pairs in each similarity bin by the number of examples in this bin. Based on cosine similarity, we estimate that out of the 332,600 pairs, the matching pairs are approximately 90,170 (27.11%). Further, based on the Jaccard distribution, we estimate that 14.79% of all tweet-conversation (root of the conversation), and 22.23% tweet–reply (the tweet before the current in the conversation) pairs are expected to match, or nearly 61,500 examples, assuming that the number of conversations and replies is equal.[6]

Our experiments show that the models can effectively account for the noise in the training data. Both the self-adaptive training and the additional weighing in the loss function (described in Section 4), gain 1 additional point MAP@5 each. These results suggest that further investigation of the potential of learning from noisy labels (Han et al., 2018; Wang et al., 2019; Song et al., 2020b,a; Zhou and Chen, 2021) and utilizing all examples in CrowdChecked, can improve the results even more. Moreover, we argue that incorporating the negative examples (non-matching pairs) from CrowdChecked in the training objective can be beneficial for the models (Lu et al., 2021; Thakur et al., 2021).

7 Conclusion and Future Work

We presented CrowdChecked, a large-scale dataset for detecting previously fact-checked claims, with more than 330,000 pairs of tweets and corresponding fact-checking articles posted by crowd fact-checkers. We further investigated two techniques for labeling the tweet–article pairs using distance supervision, resulting in training sets of 3.5K–50K examples. We also proposed an approach for training from noisy data using self-adaptive learning and additional weights in the loss function. Furthermore, we demonstrated the utility of our data, which yielded sizable performance gains of four points in terms MRR, P@1, and MAP@5 over strong baselines trained on manually annotated data (Shaar et al., 2021). Finally, we demonstrated improvements over the state of the art on the CLEF-2021 CheckThat! dataset (Chernyavskiy et al., 2021b) by two points absolute, when using CrowdChecked and our proposed model.

In future work, we plan to experiment with more languages and more distant supervision techniques such as predictions from an ensemble model.

[5] Due to the small number of annotated examples the variance in the estimates is large.

[6] In practice, there are more replies than conversations.

Ethics and Broader Impact

Dataset Collection

We collected the dataset using the Twitter API.[7] We followed the terms of use outlined by Twitter.[8] Specifically, we only downloaded public tweets, and we only distribute dehydrated Twitter IDs.

Biases

We note that some of the annotations are subjective, and we have clearly indicated in the text which these are. Thus, it is inevitable that there would be biases in our dataset. Yet, we have a very clear annotation schema and instructions, which should reduce biases.

Misuse Potential

Most datasets compiled from social media present some risk of misuse. We, therefore, ask researchers to be aware that our dataset can be maliciously used to unfairly moderate text (e.g., a tweet) that may not be malicious based on biases that may or may not be related to demographics and other information within the text. Intervention with human moderation would be required in order to ensure this does not occur.

Intended Use

Our dataset can enable automatic systems for analysis of social media content, which could be of interest to practitioners, professional fact-checker, journalists, social media platforms, and policymakers. Such systems can be used to alleviate the burden for social media moderators, but human supervision would be required for more intricate cases and in order to ensure that the system does not cause unintended harm.

Our models can help fight the COVID-19 infodemic, and they could support analysis and decision making for the public good. However, the models can also be misused by malicious actors. Therefore, we ask the potential users to be aware of potential misuse. With the possible ramifications of a highly subjective dataset, we distribute it for research purposes only, without a license for commercial use. Any biases found in the dataset are unintentional, and we do not intend to do harm to any group or individual.

[7] http://developer.twitter.com/en/docs

[8] http://developer.twitter.com/en/developer-terms/agreement-and-policy

Environmental Impact

We would like to warn that the use of large-scale Transformers requires a lot of computations and the use of GPUs/TPUs for training, which contributes to global warming (Strubell et al., 2019). This is a bit less of an issue in our case, as we do not train such models from scratch; rather, we fine-tune them on relatively small datasets. Moreover, running on a CPU for inference, once the model is fine-tuned, is perfectly feasible, and CPUs contribute much less to global warming.

References

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. 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)*, pages 4685–4697, Hong Kong, China. Association for Computational Linguistics.

Ramy Baly, Georgi Karadzhov, Jisun An, Haewoon Kwak, Yoan Dinkov, Ahmed Ali, James Glass, and Preslav Nakov. 2020. What was written vs. who read it: News media profiling using text analysis and social media context. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3364–3374, Online. Association for Computational Linguistics.

Alberto Barrón-Cedeno, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, et al. 2020. Overview of checkthat! 2020: Automatic identification and verification of claims in social media. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 215–236. Springer.

Mostafa Bouziane, Hugo Perrin, Aurélien Cluzeau, Julien Mardas, and Amine Sadeq. 2020. Team buster. ai at checkthat! 2020 insights and recommendations to improve fact-checking. In *CLEF (Working Notes)*.

Anton Chernyavskiy, Dmitry Ilvovsky, Pavel Kalinin, and Preslav Nakov. 2021a. Batch-softmax contrastive loss for pairwise sentence scoring tasks. *arXiv preprint arXiv:2110.15725*.

Anton Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. 2021b. Aschern at CLEF CheckThat! 2021: Lambda-Calculus of Fact-Checked Claims. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 484–493. CEUR-WS.org.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, pages 4171–4186, Minneapolis, Minnesota, USA.

Tamer Elsayed, Preslav Nakov, Alberto Barrón-Cedeno, Maram Hasanain, Reem Suwaileh, Giovanni Da San Martino, and Pepa Atanasova. 2019. 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*, pages 301–321. Springer.

Andreas Ferreira, William andVlachos. 2016. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1168, San Diego, California. Association for Computational Linguistics.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2021. A survey on automated fact-checking. *arXiv preprint arXiv:2108.11896*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8536–8546, Red Hook, NY, USA. Curran Associates Inc.

Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A richly annotated corpus for different tasks in automated fact-checking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning*, CoNLL '19, pages 493–503, Hong Kong, China.

Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2021. A survey on stance detection for mis-and disinformation identification. *arXiv preprint arXiv:2103.00242*.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li, and Mark Tremayne. 2017. Claimbuster: The first-ever end-to-end fact-checking system. *Proc. VLDB Endow.*, 10(12):1945–1948.

Matthew Henderson, Rami Al-Rfou, B. Strope, Yun-Hsuan Sung, L. Lukács, R. Guo, S. Kumar, B. Miklos, and R. Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *ArXiv 1705.00652*.

Lang Huang, Chao Zhang, and Hongyang Zhang. 2020. Self-adaptive training: beyond empirical risk minimization. In *Advances in Neural Information Processing Systems*, volume 33.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A survey on truth discovery. *SIGKDD Explor. Newsl.*, 17(2):1–16.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv:1907.11692*.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, TeachingNLP '02, pages 63–70, Philadelphia, Pennsylvania, USA.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Jing Lu, Gustavo Hernandez Abrego, Ji Ma, Jianmo Ni, and Yinfei Yang. 2021. Multi-stage training with improved negative contrast for neural passage retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6091–6103, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro,

and Preslav Nakov. 2020. A survey on computational propaganda detection. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4826–4832. International Joint Conferences on Artificial Intelligence Organization. Survey track.

Simona Mihaylova, Iva Borisova, Dzhovani Chemishanov, Preslav Hadzhitsanev, Momchil Hardalov, and Preslav Nakov. 2021. DIPS at CheckThat! 2021: Verified Claim Retrieval. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 558–571. CEUR-WS.org.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, NeurIPS '19, pages 8024–8035.

Dean Pomerleau and Delip Rao. 2017. Fake news challenge stage 1 (FNC-I): Stance detection. https://www.fakenewschallenge.org/.

K. Popat et al. 2016. Credibility assessment of textual claims on the web. In *CIKM*. ACM.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*.

Albert Pritzkau. 2021. NLytics at CheckThat! 2021: Multi-class fake news detection of news articles and domain identification with RoBERTa - a baseline model. In *CLEF*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Nils Reimers and Iryna Gurevych. 2019. 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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a known lie: Detecting previously fact-checked claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online. Association for Computational Linguistics.

Shaden Shaar, Fatima Haouari, Watheq Mansour, Maram Hasanain, Nikolay Babulkov, Firoj Alam, Giovanni Da San Martino, Tamer Elsayed, and Preslav Nakov. 2021. Overview of the CLEF-2021 CheckThat! Lab Task 2 on Detecting Previously Fact-Checked Claims in Tweets and Political Debates. In *CLEF (Working Notes)*, pages 393–405.

Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. 2020a. Learning from noisy labels with deep neural networks: A survey. *ArXiv*, abs/2007.08199.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2020b. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*.

Peter Stefanov, Kareem Darwish, Atanas Atanasov, and Preslav Nakov. 2020. Predicting the topical stance and political leaning of media using tweets. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 527–537, Online. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Andon Tchechmedjiev, Pavlos Fafalios, Katarina Boland, Malo Gasquet, Matthäus Zloch, Benjamin Zapilko, Stefan Dietze, and Konstantin Todorov. 2019. Claimskg: A knowledge graph of fact-checked claims. In *International Semantic Web Conference*, pages 309–324. Springer.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data augmentation method for improving bi-encoders for

pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. 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, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, NIPS '17, pages 5998–6008, Long Beach, California, USA.

Nguyen Vo and Kyumin Lee. 2019. Learning from fact-checkers: Analysis and generation of fact-checking language. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 335–344, New York, NY, USA. Association for Computing Machinery.

Nguyen Vo and Kyumin Lee. 2020. Where are the facts? searching for fact-checked information to alleviate the spread of fake news. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7717–7731, Online. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Hao Wang, Bing Liu, Chaozhuo Li, Yan Yang, and Tianrui Li. 2019. Learning with noisy labels for sentence-level sentiment classification. 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)*, pages 6286–6292, Hong Kong, China. Association for Computational Linguistics.

William Yang Wang. 2017. "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 (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. 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, pages 38–45, Online.

Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270.

Di You, Nguyen Vo, Kyumin Lee, and Qiang Liu. 2019. Attributed multi-relational attention network for fact-checking url recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1471–1480.

Wenxuan Zhou and Muhao Chen. 2021. Learning from noisy labels for entity-centric information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5381–5392, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Arkaitz Zubiaga. 2018. A longitudinal assessment of the persistence of Twitter datasets. *Journal of the Association for Information Science and Technology*, 69(8):974–984.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Comput. Surv.*, 51(2).

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016a. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PLOS ONE*, 11(3):1–29.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016b. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989.

Appendix

A Hyperparameters and Fine-Tuning

Common Parameters

- The models are developed in Python using PyTorch (Paszke et al., 2019), the Transformers library (Wolf et al., 2020) and the Sentence Transformers library (Reimers and Gurevych, 2019)[9].
- For model optimization we use AdamW (Loshchilov and Hutter, 2017) with weight decay 1e-8, β_1 0.9, β_2 0.999, ϵ 1e-08, for 10 epochs and maximum sequence length of 128 tokens (per encoder).[10]
- All SentenceBERT models are initialized from the *'stsb-bert-base'*[11] checkpoint.
- The SBERT models use cosine similarity both during training inside the MNR loss and during inference for ranking.
- The values of the hyper-parameters were selected on the development set of CheckThat '21[12] and we chose the best model checkpoint based on the performance on the development set (MAP@5).
- We ran each experiment three times with different seeds and averaged all the metrics.
- The models were evaluated on each epoch or 250 steps, whichever is less.
- The evaluation metrics are calculated using the official code from the CheckThat '21 competition (Shaar et al., 2021)[13] and the SentenceTransformer's library.
- We trained our models on 5x Tesla K80 GPUs and 1x GeForce GTX 1080Ti, depending on the dataset size, the experiments took between 10 minutes and 5 hours.

Baseline SBERT

- Baseline SentenceBERT is trained w/ LR 2e-05, warmup 0.1, and batch size 32.
- We set the temperature (τ) in the MNR loss to 1.0, i.e., using unmodified MNR.
- The model consists 110M params, same as the bert-base Devlin et al. (2019), as it uses a bi-encoder scheme.

Proposed Model

- The model is trained w/ LR 1e-05, warmup 0.1, and batch size 8, group size of 4 during the dataset shuffling.
- We tuned settings of the self-adaptive training approach: momentum α to 0.9, refurbishment process starting at the second epoch.
- We set the learning rate for temperature (τ) in the MNR loss to 0.4.
- In the re-ranking, we used 800 training examples to train SBERT and the remaining 199 to train LambdaMART.
- We re-ranked the top-100 results from the best SentenceBERT model with LambdaMART.
- All other training details we kept from (Chernyavskiy et al., 2021b).
- The model consists 330M params, 3x as the size of the Baseline SBERT, as it trains three separate models.
- In our preliminary experiments, SBERT-base and SBERT-large models achieved the same results in terms of MAP@5, therefore we experiment with the *base* versions.

B Dataset

B.1 Tweet Collection (Conversation Structure)

It is important to note that this *'fact-checking'* tweet can be part of a multiple-turn conversational thread, therefore taking the post that it replies to (previous turn), does not always express a claim which the current tweet targets. In order to better understand that phenomena, we perform manual analysis of conversation thread. The conversational threads are organized in a similar way shown Figure 1, i.e., the root is the first comment, then there can be a long discussion, followed by a fact-checking comment (the one with the Snopes link). In our analysis we identify four patterns: (*i*) current tweet verifies a claim in the the tweet it replies to, (*ii*) the tweet verifies the root of the conversation, (*iii*) the tweet does not verify any claim in the chain (a common scenario), (*iv*) in very few cases the fact-check targets a claim expressed not in the root or the closest tweet. This analysis suggests that for the task of detecting previously fact-checked claims, it is sufficient to collect the triplet of the fact-checking tweet, root of the conversation (*conversation*), and the tweet that the target tweet is replying to (reply).

[9] github.com/UKPLab/sentence-transformers

[10] When needed, we truncated the sequences token by token, starting from the longest sequence in the pair.

[11] huggingface.co/sentence-transformers/stsb-bert-base

[12] https://gitlab.com/checkthat_lab/clef2021-checkthat-lab/-/tree/master/task2

[13] https://gitlab.com/checkthat_lab/clef2021-checkthat-lab/-/tree/master/task2/scorer

B.2 Fact-checking Articles Collection

In order to obtain a collection of fact-checking articles for each tweet, we first formed a list of unique URLs shared in the fact-checking tweets from the crowd fact-checkers. Next, from each URL we downloaded the HTML of the whole page and extracted the meta information using CSS selectors and RegEx rules. In particular, we follow previous work (Barrón-Cedeno et al., 2020; Shaar et al., 2021) and collect: *title*, the title of the page, *subtitle*, short description of the fact-check, *claim*, the claim of interest, *subtitle*, short description of the fact-check, *date*, the date on the article was published, *author*, the author of the article. We do not parse the content of the article and factual label, as the credibility of the claim is not related to the objective of this task, i.e., the goal is to find a fact-checking article, but not to verify it.

As a result we collected 10,340 articles that are published in the period between 1995–2021. The per-year distribution is shown in Table 2 (in brown). The majority of the articles are from the period after 2015, with a peak at the ones from 2020/2021. We attribute this on the increased media literacy and on the nature of the Twitter dynamics (Zubiaga, 2018).

B.3 CheckThat '21 Word Overlaps

Here, we analyze the distribution of the Jaccard scores in the CheckThat '21, shown in Figure 4. The distribution is different compared to the one observed in the our newly collected dataset, as it peaks at around 0.4, and is slightly shifted towards lower similarity values, suggesting the examples included are not easily solvable with basic lexical features (Shaar et al., 2021), which we also observe in our experiments (see Section 5).

C Annotations

Setup and Guidelines Each annotator was provided by the same guidelines and briefed in from one of the authors of this paper. For annotation we used a Google Sheets document, where non of the annotators had access to the annotations from the others. The annotation sheet contained the following fields:

- *tweet_text* – the text of the fact-checking tweet
- *text_conversation* – the text of the root of the conversation
- *text_reply* – the text of the last tweet before the fact-checking one

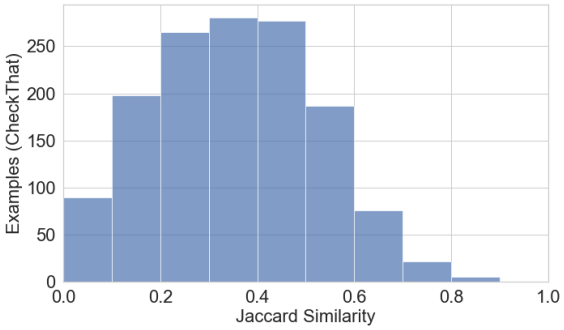

Figure 4: Distribution of the Jaccard similarity scores. The score is an average of the *sim(tweet, title)* and *sim(tweet, subtitle)*.

	Replay	Conversation
Fleiss Kappa	0.745	0.750

Table 7: Fleiss Kappa inter-annotator agreement between our three annotators (A, B, C).

- *title* – the title of the Snopes article
- *subtitle* – the subtitle of the Snopes article.

The task annotation task is to mark if 'Conversation matches' and 'Replay matches' using a checkboxes. We also allowed them to add comments as a free form text.

Demographics We recruited three annotators – 2 male and 1 female on age between 25 and 30. The annotators have higher education (at least a bachelors degree), and are currently enrolled in a MSc or Ph.D programs in computer science. Each annotator is proficient in English but is not a native speaker of the language.

Inter-annotator Agreement Here, we present the inter-annotator agreement. We measure the overall agreement using Fleiss kappa (Fleiss, 1971) (shown in Figure 7) but also the agreement between each two annotators using Cohen's Kappa (shown in Table 8). The overall level of agreement of agreement between the annotators is good. Moreover, we can see that between annotator A and C the agreement is almost perfect both for the replies and conversations. The lowest agreement is between A and B but still substantial.

Disagreement Analysis After the annotations procedure was finished we analyzed the examples the annotators disagree on. The first type of claims that cause disagreement are the ones depend on information external sources, e.g., *'Blame Russia*

Annotators	Replay	Conversation
	Cohen Kappa	
A ↔ B	0.650	0.655
A ↔ C	0.885	0.922
B ↔ C	0.698	0.673

Table 8: Cohen Kappa inter-annotator agreement between our three annotators (A, B, C).

again? [URL]'. The second type are tweets containing multiple claims that needs to be fact-checked, however the referenced article does not target the main claim, e.g., *'It sounds like someone who is scared as heck that they will not win," Shermichael Singleton says of Pres. Trump's remarks encouraging his supporters to vote twice.'* and its crowd fact-check *'Did Trump Tell People To Vote Twice?'*. Here, the main claim is in the quote itself, while the remark about voting twice is secondary. Third type are the claim is ambiguous *Fanta (soft drink) was created so that the Nazi's could replace Coca-Cola during WWII [URL]*, and the fact-check is about *'Was Fanta invented by the Nazis?'*. Here, it is not clear who created Fanta. The final pattern is – the claim is partial match with the fact-check, e.g., *'did President Trump have a great economy and job creation for 1st 3 years???'*, and the fact-check is *'Did Obama's Last 3 Years See More Jobs Created Than Trump's First 3?'*.

D Experiments

D.1 Baselines and State-of-the-Art

Retrieval (Shaar et al., 2021) uses an information retrieval model based on BM25 (Robertson and Zaragoza, 2009) that ranks the list of fact-checking articles based on the relevance score between its *{'claim', 'title'}* and the tweet's text.

Sentence-BERT is a bi-encoder model based on Sentence-BERT fine-tuned for detecting previously fact-checked claims using MNR loss. The details are in Section 4, *General Scheme*.

DIPS (Mihaylova et al., 2021) adopts a Sentence-BERT model that computes the cosine similarity for each pair of an input tweet and a verified claim (article). The final ranking is made by passing a sorted list of cosine similarities to a fully-connected neural network.

NLytics (Pritzkau, 2021) uses a RoBERTa-based model optimized as a regression function obtaining a direct ranking for each tweet-article pair.

Aschern (Chernyavskiy et al., 2021b) combines TF.IDF with a Sentence-BERT (ensemble with three models of each type). The final ranking is obtained from a re-ranking LambdaMART model.

Model	MRR	P@1	MAP@5
Baselines (CheckThat '21)			
Retrieval (Shaar et al., 2021)	76.1	70.3	74.9
SBERT (CheckThat '21)	87.97	84.92	87.45
CrowdChecked (Our Dataset)			
SBERT (cos > 0.50)	88.20	85.76	87.80
SBERT (cos > 0.60)	87.21	84.25	86.69
SBERT (cos > 0.70)	86.18	83.08	85.76
SBERT (cos > 0.80)	83.57	80.40	82.93
SBERT (jac > 0.30)	88.01	85.09	87.61
SBERT (jac > 0.40)	87.26	84.76	86.80
SBERT (jac > 0.50)	86.53	83.42	86.13
(Pre-train) CrowdChecked, (Fine-tune) CheckThat '21			
SBERT (cos > 0.50, Seq)	89.92	87.60	89.49
SBERT (cos > 0.60, Seq)	89.56	87.27	89.20
SBERT (cos > 0.70, Seq)	88.70	85.59	88.36
SBERT (cos > 0.80, Seq)	88.42	85.26	88.03
SBERT (jac > 0.30, Seq)	90.21	87.44	89.69
SBERT (jac > 0.40, Seq)	89.64	86.77	89.25
SBERT (jac > 0.50, Seq)	89.44	86.26	89.03
(Mix) CrowdChecked and CheckThat '21			
SBERT (cos > 0.50, Mix)	89.47	86.77	88.99
SBERT (cos > 0.60, Mix)	88.54	85.76	87.98
SBERT (cos > 0.70, Mix)	87.71	84.92	87.18
SBERT (cos > 0.80, Mix)	88.40	85.26	87.97
SBERT (jac > 0.30, Mix)	90.41	87.94	90.00
SBERT (jac > 0.40, Mix)	89.82	86.60	89.48
SBERT (jac > 0.50, Mix)	88.71	85.26	88.31

Table 9: Evaluation on the CheckThat '21 **development** set. In parenthesis is name the training split, i.e., Jaccard (*jac*) or Cosine (*cos*) data selection strategy, *(Seq)* first training on CrowdChecked and then on CheckThat '21, *(Mix)* mixing the data from the two datasets.

D.2 Experimental Results

Here, we present the expanded results for our experiments described in Section 5. Tables 9 and 10 include the results for the *Data Efficiency* experiments on the development set, and testing set, respectively. In Table 10 corresponds to Table 5 in the main paper, and includes all metrics and for all thresholds (shown in Figure 3). Next, the results from our *Modeling Noisy Data* experiments are in Table 11, which corresponds to Table 6 in the main paper. In all tables we use the same notation and grouping as in their corresponding table.

Model	MRR	Precision @1	Precision @3	Precision @5	Precision @10	Precision @20	MAP @1	MAP @3	MAP @5	MAP @10	MAP @20
Baselines (CheckThat '21)											
Retrieval (Shaar et al., 2021)	76.1	70.3	26.2	16.4	8.8	4.6	70.3	74.1	74.9	75.7	75.9
SBERT (CheckThat '21)	79.96	74.59	27.89	17.19	8.96	4.61	74.59	78.66	79.20	79.66	79.83
CrowdChecked (Our Dataset)											
SBERT (cos > 0.50)	81.58	75.91	28.60	17.76	9.04	4.67	75.91	80.36	81.05	81.27	81.48
SBERT (cos > 0.60)	79.71	74.75	27.39	16.96	8.86	4.59	74.75	78.25	78.84	79.38	79.61
SBERT (cos > 0.70)	78.27	72.28	27.61	17.10	8.89	4.53	72.28	76.95	77.54	78.01	78.12
SBERT (cos > 0.80)	78.39	72.94	27.34	16.83	8.81	4.55	72.94	77.04	77.52	78.08	78.28
SBERT (jac > 30)	81.50	76.40	28.49	17.43	8.94	4.65	76.40	80.45	80.84	81.14	81.38
SBERT (jac > 40)	79.45	74.42	27.34	16.93	8.89	4.65	74.42	77.92	78.52	79.08	79.33
SBERT (jac > 50)	79.96	74.75	27.89	17.29	8.94	4.60	74.75	78.63	79.26	79.63	79.81
(Pre-train) CrowdChecked, (Fine-tune) CheckThat '21											
SBERT (cos > 0.50, Seq)	82.26	77.06	28.27	17.62	9.26	4.76	77.06	80.64	81.41	81.99	82.18
SBERT (cos > 0.60, Seq)	80.13	75.41	27.45	17.00	8.94	4.65	75.41	78.55	79.13	79.76	79.99
SBERT (cos > 0.70, Seq)	79.27	73.43	27.72	17.33	8.94	4.58	73.43	77.78	78.56	78.94	79.09
SBERT (cos > 0.80, Seq)	78.32	72.77	27.17	16.93	8.89	4.58	72.77	76.71	77.41	77.98	78.15
SBERT (jac > 0.30, Seq)	83.76	78.88	28.93	17.82	9.21	4.71	78.88	82.59	83.11	83.49	83.63
SBERT (jac > 0.40, Seq)	80.69	75.25	27.83	17.33	9.09	4.69	75.25	79.04	79.76	80.34	80.57
SBERT (jac > 0.50, Seq)	81.99	76.90	28.16	17.76	9.13	4.69	76.90	80.34	81.33	81.70	81.88
(Mix) CrowdChecked and CheckThat '21											
SBERT (cos > 0.50, Mix)	82.12	76.57	28.55	17.59	9.13	4.68	76.57	80.86	81.38	81.82	82.00
SBERT (cos > 0.60, Mix)	81.45	76.40	28.27	17.43	8.96	4.61	76.40	80.25	80.79	81.14	81.31
SBERT (cos > 0.70, Mix)	79.08	73.10	27.83	17.33	8.89	4.57	73.10	77.72	78.46	78.77	78.95
SBERT (cos > 0.80, Mix)	79.73	74.75	27.56	17.00	9.06	4.62	74.75	78.22	78.73	79.46	79.59
SBERT (jac > 0.30, Mix)	83.04	78.55	28.66	17.52	9.11	4.69	78.55	81.93	82.30	82.75	82.94
SBERT (jac > 0.40, Mix)	81.18	74.59	28.55	17.72	9.14	4.74	74.59	79.79	80.46	80.85	81.10
SBERT (jac > 0.50, Mix)	81.56	76.73	28.22	17.36	9.03	4.71	76.73	80.23	80.71	81.19	81.45

Table 10: Evaluation on the CheckThat '21 **testing** set. In parenthesis is name the training split, i.e., Jaccard (*jac*) or Cosine (*cos*) data selection strategy, *(Seq)* first training on CrowdChecked and then on CheckThat '21, *(Mix)* mixing the data from the two datasets.

Model	MRR	Precision @1	Precision @3	Precision @5	Precision @10	MAP @1	MAP @3	MAP @5	MAP @10
DIPS (Mihaylova et al., 2021)	79.5	72.8	28.2	17.7	9.2	72.8	77.8	78.7	79.1
NLytics (Pritzkau, 2021)	80.7	73.8	28.9	17.9	9.3	73.8	79.2	79.9	80.4
Aschern (Chernyavskiy et al., 2021b)	88.4	86.1	30.0	18.2	9.2	86.1	88.0	88.3	88.4
SBERT (jac > 0.30, Mix)	83.0	78.6	28.7	17.5	9.1	78.6	81.9	82.3	82.8
+ shuffling & trainable temp.	83.2	77.7	29.1	17.8	9.1	77.7	82.2	82.6	82.9
+ self-adaptive training (Eq. 1)	84.2	78.7	29.3	18.1	9.3	78.7	83.0	83.6	83.9
+ loss weights	84.8	79.7	29.5	18.2	9.3	79.7	83.7	84.3	84.6
+ TF.IDF + Re-ranking	89.9	86.1	30.9	18.9	9.6	86.1	89.2	89.7	89.8
+ TF.IDF + Re-ranking (ens.)	90.6	87.6	30.7	18.8	9.5	87.6	89.9	90.3	90.4

Table 11: Results on the CheckThat '21 **testing** set. We compare our model and its components (added sequentially) to state-of-the-art approaches.