University of Copenhagen Participation in TREC Health Misinformation Track 2020

Lucas Chaves Lima[*], Dustin Brandon Wright[*], Isabelle Augenstein, and Maria Maistro

University of Copenhagen
{*lcl,dw,augenstein,mm*}*@di.ku.dk*

Abstract

In this paper, we describe our participation in the TREC Health Misinformation Track 2020. We submitted 11 runs to the Total Recall Task and 13 runs to the Ad Hoc task. Our approach consists of 3 steps: (1) we create an initial run with BM25 and RM3; (2) we estimate credibility and misinformation scores for the documents in the initial run; (3) we merge the relevance, credibility and misinformation scores to re-rank documents in the initial run. To estimate credibility scores, we implement a classifier which exploits features based on the content and the popularity of a document. To compute the misinformation score, we apply a stance detection approach with a pretrained Transformer language model. Finally, we use different approaches to merge scores: weighted average, the distance among score vectors and rank fusion.

1 Introduction

The spread of fake news and misinformation has become a serious issue affecting society in many ways. It can, for instance, influence public opinion to promote a political candidate and substantially affect the final election outcome. It can also have deleterious effects on peoples' reputation, especially on social networks. When it comes to health, it can harm people by guiding them to take wrong decisions, which, in the worst cases, can lead people to them injure themselves. During the COVID-19 pandemic, we all witness how misinformation can have dangerous and have severe consequences on peoples' choices and lives.

In this context, Information Retrieval (IR) systems have a central position, as they can play an active role in contrasting the spread of misinformation [17]. Documents which are relevant, credible and *incorrect* represent a serious threat for users, and should be discarded or presented very low in rankings [18]. Therefore, there is a need to design IR systems that can promote relevant, credible and correct information over incorrect information.

This paper presents out attempt to address this challenging problem through our system submitted to the TREC Health Misinformation Track 2020. The Track offers 2 tasks: the goal of the *Total Recall* task is to identify all the documents which convey misinformation, while the goal of the *Ad Hoc* task is to promote documents which convey relevant, correct and credible information.

We design a solution which consists of 3 main steps. First, we generate an initial ranking of 1 000 documents with BM25 and RM3 [16]. This initial ranking accounts for only relevance and serves solely as an initial filter to identify a smaller set of possibly relevant documents. The subsequent step involves the estimation of credibility and misinformation scores. We compute them in an independent way and adjust for documents retrieved in the initial run.

To estimate the credibility score, we design a classifier which combines the predictions of 4 different models. As features, we use both content features (i.e. number of CSS definitions and readability) and social features (i.e. features based on page rank). The classifier is trained on the Microsoft Credibility dataset [27].

[*] The first and second author contributed equally.

To estimate the misinformation score, we use a popular approach from the fact checking domain. First, we use a stance detection [21] model to identify the stance of a topic inside a document, then, we determine whether the stance agrees with the topic or not. As the stance detection model we use a Roberta [19] based classifier trained on the Fake News Challenge-1 dataset.[1]

As last step, we need to merge the relevance, credibility and misinformation scores to re-rank documents in the initial run. Since all scores result from different systems, we first normalize them as z_scores. Then, we implement 3 different strategies to merge those scores: (1) we consider the weighted average; (2) we consider the best score for each aspect and compute the distance between documents scores and the best scores; (3) we use reciprocal rank fusion [10].

The paper is organized as follows: Section 2 provides a brief description of the Health Misinformation Track; Section 3 describes our approach in details; Section 4 reports the performance of our runs and discusses the experimental results; Section 6 presents conclusions and future work.

2 Track Description

The TREC Health Misinformation Track 2020 includes two different tasks, the Total Recall task and the Ad Hoc task, presented in Section 2.1. Both tasks use the CommonCrawl News crawl[2], which is described in Section 2.2. An accurate description of the tasks can be found in the track overview paper [9].

2.1 Tasks

The goal of the Health Misinformation Track is to design and develop IR Systems able not only to retrieve relevant documents, but also to rank credible and correct documents before not credible and incorrect documents. The Total Recall and Ad Hoc tasks are mirroring each other: the purpose of the Total Recall task is to retrieve all documents that are useful and incorrect, thus *harmful* documents that convey misinformation; the purpose of the Ad Hoc task is to rank *helpful* documents, i.e. useful, credible, and correct documents, above harmful documents.

2.2 Test Collection

The corpus of the Total Recall and Ad Hoc Tasks consits of news articles in the CommonCrawl News crawl, sampled from January, 1st 2020 to April 30th, 2020. The dataset includes articles in different languages, but non English documents are considered not useful.

The tasks provide 49 topics (officially 50, but topics 7 and 17 are repeated). All topics are about health and COVID-19. Each topic has a title, a description, which reformulates the title as a question, a yes/no answer, which is the actual answer to the description field based on the provided evidence, and a narrative, to describe helpful and harmful documents in relation to the given topic.

Documents were judged with respect to three aspects: *usefulness*, i.e. whether a document contains useful information to answer the question in the topic description field, *credibility*, i.e. whether a document is considered credible by the assessor, and *correctness*, i.e. whether a document answers to the topic question as specified in the answer field.

3 Approach

In this section, we present our approach for the Total Recall and Ad Hoc tasks. For both tasks, the challenge is to rank documents not solely by their usefulness score, but also by their credibility and correctness scores. We estimate usefulness with credibility, i.e. we use basic IR models to retrieve relevant documents. Similarly, we estimate correctness by computing the misinformation score with a fact checking approach.

Since it was not computationally feasible to estimate a relevance, credibility and misinformation score for each document in the collection, we break down the problem as follows:

[1] http://www.fakenewschallenge.org/

[2] https://commoncrawl.org/2016/10/news-dataset-available/

1. We exploit the entire collection and produce an initial run R based on relevance only;
2. We estimate credibility and misinformation scores for each document $d \in R$ and for each topic q;
3. We re-rank documents in R based on relevance, credibility and misinformation.

In the following, Section 3.1 presents how we build the the initial run based on relevance, Section 3.2 describes how we compute credibility scores, Section 3.3 details how we compute misinformation scores and finally Section 3.4 explains how we merged relevance, credibility and misinformation scores to re-rank documents and produce the final run.

3.1 Estimating Relevance

We select two simple unsupervised models to generate the initial run R: BM25 and a language model combined with pseudo-relevance feedback RM3 [16]. To retrieve documents, we use the topic title and description as query. For each topic, we retrieve the top 1 000 documents, which became the candidate initial run R.

To create the index and implement the retrieval models, we use the already available implementations by Anserini[3], which is a java toolkit built over the open-source search library Lucene[4]. To build the index, we remove standard English stop words[5] and use Porter stemmer.

BM25 has two parameters b and $k1$, which represent a correction factor for document length normalization and control term frequency saturation respectively. We set $b = 0.4$ and $k1 = 0.9$, which is the default Anserini configuration. For RM3 model, we also use the default Anserini configuration, which is $fb_{terms} = 10, fb_{docs} = 10$ and original_query_weight = 0.5.

3.2 Estimating Credibility

We frame the credibility estimation prediction as a classification problem. Formally, consider a fixed set of credibility classes $L = \{l_1, l_2, \ldots, l_n\}$, representing the credibility labels for a document, for example $L = \{1, 2, 3, 4, 5\}$, where the higher the integer, the higher the credibility. Each document d is represented by a feature vector $\vec{x} \in X$. Given a training set of m labeled documents $\{(\vec{x}_1, l_1), \ldots, (\vec{x}_m, l_m)\}$, where $\vec{x}_i$ represents the feature vector of the i^{th} training document d_i and l_i its class label, the classifier fits a function $\gamma : X \to L$, such that for a new unseen document d_k and its features $\vec{x}_k$, the classifier can predict $l_k \in L$.

We consider credibility as a topic independent property of documents, meaning that its estimation can be based solely on the document, disregarding which topic the document was retrieved for. Therefore, we propose a classification algorithm which approximates the credibility of a document based on its textual and design content, and publicly available information estimating documents popularity.

Credibility features

Inspired by Olteanu et al. [24], we devise our feature set to represent documents. These features are summarized in Table 1 and organized into two groups as suggested in [24]: Content Features and Social Features.

Content Features are extracted from the raw HTML content of the document. They consist of features based on the Web page textual content (i.e. text readability), or based on the Web page structure (i.e. number of CSS definitions);

Social Features Publicly available information about the Web page can be a good indicator of its credibility. For instance, the popularity of a document can be approximated by its PageRank. The underlying assumption is that, incoming links to a Web page can be seen as endorsements for the Web page and, thus, they could be a good indicator for a Web page popularity [24]. We assume that popular pages are also reliable.

[3] https://github.com/castorini/Anserini
[4] https://lucene.apache.org
[5] Stopwords list.
[6] https://pypi.org/project/textstat/
[7] https://www.domcop.com/openpagerank/

Table 1: Set of features used for credibility detection.

Content Features	
css_definitions	# of CSS definitions in the raw HTML content of the document
text_readability	Estimated school grade level required to understand the text extracted using textstat[6]
Social Features	
pr_rank	The rank position of a document based on page rank extracted using OpenPageRank API[7]
page_rank_integer	The rounded page rank score of the document extracted using OpenPageRank API
page_rank_decimal	The page rank score of the document extracted using OpenPageRank API
toplevel_domain	Check weather the Web page URL contains .edu, .gov, .com, etc.

Classification Models

As the credibility model, we use four different state-of-the-art classification approaches implemented with the library scikit-learn [8]: Logistic Regression, Random Forest, SVM Logistic, and Naive Bayes. We combine the predictions of the classifiers using a soft VotingClassifier, which predicts the class label based on the argmax of the sums of the predicted probabilities. For all the different classification approaches we used the default parameters of the scikit-learn library.

Training

Since there are no credibility assessments for the Common Crawl News sample used for this track, we could not train our model on this corpus. We instead use the well known Microsoft Credibility dataset [27], which is a Web dataset not only containing health-related credibility assessments, but also more general topics (e.g., celebrities, politics). This dataset consists of a set of Web pages retrieved for different queries. For each retrieved Web page, it contains its URL, rank position, and a credibility label in $L = \{1, 2, 3, 4, 5\}$.

To simplify the task, we consider a binary classifier. Therefore, we map credibility labels to two classes as follows: labels $1, 2, 3$ are mapped to 0 and $4, 5$ are mapped to 1.

3.3 Estimating Misinformation

We frame the estimation of misinformation as a stance detection problem. The goal of stance detection is: given a claim C and a statement S, determine if S agrees with, disagrees with, or is neutral towards C. In this, stance detection is generally approached as a supervised learning problem with labelled pairs (C,S) [21, 4], and is similar to the problem setting in other fact checking tasks e.g. FEVER [31].

We start with the set of topics (queries) and answers provided for the task $\{(t_1, a_1), (t_2, a_2), ..., (t_{50}, t_{50})\} \in M$, with $a_k \in \{0, 1\}$ corresponding to "no" and "yes," and corpus of articles after the initial run R. Each (t_k, a_k) tuple is paired with a subset of documents $\hat{R}_k \subset R$ using one of the ranking approaches described in §3.1. The goal is then to predict the stance of each document $d_k^{(i)} \in \hat{R}_k$ towards the topic t_k, and then rank each document based on the relationship of the predicted stance towards the answer a_k.

More concretely, a stance prediction model takes as input the topic t_k and a document $d_k^{(i)}$, and produces probabilities $\mathbb{P}(l|t_k, d_k^{(i)})$ for labels $l \in \{0, 1, 2\}$ corresponding to levels of disagreement, agreement, and neutrality. The final misinformation score s then takes into account how much the model predicts the document both disagrees and agrees with the topic via a subtractive measure.

$$s(k, i) = \mathbb{P}\left(1 - a_k|t_k, d_k^{(i)}\right) - \mathbb{P}\left(a_k|t_k, d_k^{(i)}\right) \tag{1}$$

The effect of this scoring function is as follows: if the answer of a given topic is "no", then the score will take the probability that the article *agrees* with the topic and subtract from this the probability that it *disagrees* with the topic. If the answer of a given topic is "yes", then the score will take the probability that the article *disagrees* with the topic and subtract from this the probability that it *agrees* with the topic. The

[8] `https://scikit-learn.org/stable/`

purpose of using the subtractive measure is to take into consideration articles which are neutral towards the topic. The net effect is that articles which are more likely to disagree with the answer will have higher scores than articles which are more likely to agree with the answer.

Training

We use a large pretrained transformer language model as our stance detection model, namely Roberta [19] from the HuggingFace transformers library [34]. Roberta is a large transformer model pretrained on massive unlabelled text corpora using a masked language modeling objective. Models trained in this manner require only to be fine-tuned on downstream tasks, and have shown to achieve state of the art performance after doing so [11]. In this, we only require a labelled dataset on which to fine-tune the model.

As no labelled dataset currently exists for the specific problem of health related stance detection within the Common Crawl News corpus, a large general domain dataset is used to train the stance detection model. In this, we use the Fake News Challenge-1 dataset[9], which consists of full text news articles, headlines, and stance labels between them. The labels can be one of {unrelated, discuss, agree, disagree}. This dataset is unique in that it is designed for performing stance detection on full text articles, making it a good fit as a source of training data. We follow previous best practice when working with this dataset and train a separate model for the binary task of predicting whether or not the headline is related to the article, and a separate model for predicting the stance, achieving comparable results to previous work on the specific task of stance detection (73.5 macro F1 score) [29].

Prior to ranking articles using the stance prediction model, we preprocess the topics by manually converting each topic description from a question to a statement in order to better match the style of data the model was trained on. In addition, the articles in the dataset tend to have a large amount of irrelevant information as a result of imperfect cleaning of the data, while the Roberta model has a limited token capacity of 512, meaning we can only perform prediction using subsets of large articles. Given this, as a preprocessing step we filter the article content by locating the first occurrence of the topic string in the description e.g. "ibuprofen," and perform misinformation ranking on the article starting from the sentence containing this string.

3.4 Merging Scores

In this section, we explain how we re-rank documents by considering relevance, credibility and misinformation scores. Observe that all the scores result as output of different models, thus, they are not directly comparable. Therefore we standardize the scores by computing per topic z-scores as follows:

$$z\_score_a(d,t) = \frac{x_a(d,t) - \mu_a(t)}{\sigma_a(t)} \tag{2}$$

where $a \in A$ refers to the aspect $A = \{\mathit{relevance}, \mathit{credibility}, \mathit{misinformation}\}$, $x_a(d,t)$ is the relevance, credibility or misinformation score of document d for topic t, $\mu_a(t)$ and $\sigma_a(t)$ are the mean and standard deviation of relevance, credibility or misinformation scores for all documents retrieved for topic t.

We use three different approaches to combine relevance, credibility and misinformation scores in a single score: weighted average, distance from the best and ranking fusion. The final single score is used to re-rank documents and produce the final run which accounts for all the three aspects.

Weighted Average

For a fixed topic t, we compute the weighted sum of z-scores for each document $d \in R$ as follows:

$$S(d,t) = \sum_{a \in A} w_a \times z\_score_a(d,t) \tag{3}$$

where w_a is the weight associated to aspect $a \in A$ (e.g. weight given to relevance score). We arbitrarily set the weights w_a to be uniform or we assign more weight to one aspect over the others, depending on the run.

[9] `https://github.com/FakeNewsChallenge/fnc-1`

Distance from the Best Score

Given a topic t, we consider the highest z_score across aspects as a reference score. More formally, let $z\_score^{\star}_{a}(t) = \max\{z\_score^{\star}_{a}(d,t) | d \in R\}$ be the highest z_score for a given topic and aspect. We define the *best score vector* as the vector whose coordinates are $z\_score^{\star}_{a}(t)$:

$$z\_score^{\star}(t) = [z\_score^{\star}_{rel}(t), z\_score^{\star}_{cred}(t), z\_score^{\star}_{mis}(t)] \tag{4}$$

Similarly, each document can be represented as a vector of z_scores. Therefore, we define the final ordering of documents by computing the distance of each document vector from the best score vector:

$$S(d,t) = dist(z\_score^{\star}(t), z\_score(d,t)) \tag{5}$$

where $z\_score(d,t)$ denotes the score vector for document d with respect to topic t. The closer the document vector to the best score vector, the higher a document should be in the ranking. We instantiate *dist* with the well-known distances, Euclidean and Chebyshev.

Observe that Equation (4) can be modified by taking the minimum instead of the maximum for certain aspects. For example, for the Total Recall task the purpose is to find documents that disagree with the answer, thus we take the maximum of the misinformation score, while for the Ad Hoc task the purpose is to find documents that agree with the answer, thus we take the minimum of the misinformation score.

Rank Fusion

We consider 3 different runs R_a by ranking documents solely based on a given aspect a. We merge these 3 runs into a single runs with Reciprocal Rank Fusion (RRF) [10]. Specifically, for each topic t, we compute the fused score of each document as follows:

$$RRF\_score(d,t) = \sum_{a \in A} \frac{1}{k + \mathrm{rank}_a(d,t)} \tag{6}$$

where $\mathrm{rank}_a(d,t)$ is the rank position of document d, when ranked with respect to topic t and aspect a; k is a parameter to control the impact of bad ranking systems, which can retrieve bad documents at high positions in the ranking. We set $k = 60$, the value used in the original paper by Cormack et al. [10].

4 Experimental Evaluation

4.1 Submitted Runs

Total Recall Task

We submitted a total of 11 runs to the Total Recall Task, described in Table 2. We combine different cut-offs for re-ranking and different merging approaches. We also create some runs by exploiting just the credibility or misinformation scores, specifically `run4`, `run5` and `run9`.

Observe that for all runs, we used the reversed credibility (i.e. by multiplying the score $\times -1$), so that low credibility documents are placed towards the beginning of the ranking. Indeed, for the Total Recall task we want the documents with low credibility, but high misinformation and relevance score, to be retrieved at the top of the ranking.

Ad Hoc Task

We submitted a total of 13 runs to the Ad Hoc Task, described in Table 3. First, we submitted 2 baseline runs: `adhoc_run1` and `adhoc_run2`. These can be seen as runs where documents are sorted just by their relevance score. As for the Total Recall Task, we submitted some runs that exploit just the credibility or misinformation score, `adhoc_run9`, `adhoc_run10`, `adhoc_run12`, `adhoc_run13`. Furthermore, we combine different cut-offs for re-ranking and different merging approaches.

Observe that we use the reversed misinformation score, since a high misinformation score means a high disagreement between the document and the topic answer. Therefore, those documents should be placed close to the bottom of the ranking.

Table 2: Runs submitted to the Total Recall Task.

RUNID	Initial Run	Re-ranking
`run1`	BM25_description	top 200 documents using weighted average $w_{rel} = -w_{cred} = w_{mis}$
`run2`	RM3_description	top 100 documents using weighted average $-w_{cred} = w_{mis}$
`run3`	RM3_description	top 100 documents using weighted average $-w_{rel} = -w_{cred} = w_{mis}$
`run4`	RM3_description	top 100 documents using only the reversed credibility score
`run5`	RM3_description	top 100 documents using only the misinformation score
`run6`	-	Reciprocal rank fusion of `run4` and `run5`
`run7`	RM3_description	top 100 documents using the Euclidean distance from the best score (min cre score)
`run8`	RM3_description	top 100 documents using the Chebyshev distance from the best score (min cre score)
`run9`	RM3_description	top 200 documents using only the reversed credibility score
`run10`	RM3_description	top 300 documents using the Euclidean distance from the best score (min cre score)
`run11`	-	Reciprocal rank fusion of all runs

Table 3: Runs submitted to the Ad Hoc Task.

RUNID	Initial Run	Re-ranking
`adhoc_run1`	BM25_description	baseline, no re-ranking
`adhoc_run2`	RM3_description	baseline, no re-ranking
`adhoc_run3`	BM25_description	top 100 documents using weighted average $w_{rel} = w_{cred} = -w_{mis}$
`adhoc_run4`	RM3_description	top 100 documents using weighted average $w_{rel} = w_{cred} = -w_{mis}$
`adhoc_run5`	BM25_description	top 100 documents using weighted average $w_{rel} = 0.6$ and $w_{cred} = -w_{mis} = 0.2$
`adhoc_run6`	RM3_description	top 100 documents using weighted average $w_{rel} = 0.6$ and $w_{cred} = -w_{mis} = 0.2$
`adhoc_run7`	RM3_description	top 100 documents using the Euclidean distance from the best score (min mis score)
`adhoc_run8`	RM3_description	top 100 documents using the Chebyshev distance from the best score (min mis score)
`adhoc_run9`	RM3_description	top 100 documents using only the credibility score
`adhoc_run10`	RM3_description	top 100 documents using only the reversed misinformation score
`adhoc_run11`	-	Reciprocal rank fusion of `adhoc_run1`, `adhoc_run9` and `adhoc_run10`
`adhoc_run12`	RM3_description	top 200 documents using only the credibility score
`adhoc_run13`	RM3_description	top 200 documents using only the reversed misinformation score

4.2 Results

Total Recall Task

Table 4 reports Rprec scores for our runs submitted at the Total Recall Task: 9 out of 11 runs are above the task median. However, Rprec scores are overall quite low, our best score is 0.1303, showing that finding useful but not correct documents is a challenging task.

Our best run is `run5`, which re-rank documents solely with the misinformation score. This is a promising results, suggesting that our approach to estimate misinformation goes towards the correct direction. Among the approaches to merge scores, rank fusion seems the most effective, followed by the Euclidean distance from the best score.

Our worst run is `run9`, which re-rank the first 200 documents by their reversed credibility score. Clearly, this strategy is too harsh, since it places close to the top of the ranking documents that are not relevant, thus not credible and not correct. The second-to-last run is `run1`, which is below the task median. This run re-ranks the top 200 documents by merging the scores with a weighted average with uniform weights. We hypothesise that this is due to the reversed credibility score, indeed `run4` re-ranks documents solely with the reversed credibility score and is the third-to-last run, just above the task median. It might be that a low credibility score denotes very low quality documents, which represent spam, and thus are not relevant.

Table 4: Total Recall Task: our runs compared to the overall median results of submitted runs.

RUNID	Rprec
run9	0.0866
run1	0.0936
median (TREC)	0.0976
run4	0.1090
run3	0.1115
run10	0.1142
run6	0.1173
run8	0.1226
run2	0.1237
run7	0.1267
run11	0.1274
run5	0.1303

Ad Hoc Task

Table 5 and Table 6 report the results for our runs in the Ad Hoc task. For all measures, except for harmfulness, our runs are above the task median. On the other side, for harmfulness, all our runs are worse than the task median, meaning that we need to improve our strategy to move harmful documents towards the end of the ranking.

Table 6 shows that our best runs always improve over our internal baselines (`adhoc_run1` and `adhoc_run2`). This is always true, except for the case of 2 aspects, correctness and credibility, evaluated with Convex Aggregating Measure (CAM) instantiated with Average Precision (AP) (Table 6, column 1, id = 0).

Figure 1 shows the performance of our runs when evaluated with Normalized Discounted Cumulated Gain (nDCG) with respect to binary multi-aspect labels: for example, in Figure 1b a document is mapped to 1 if it is useful and credible, independently of its correctness.

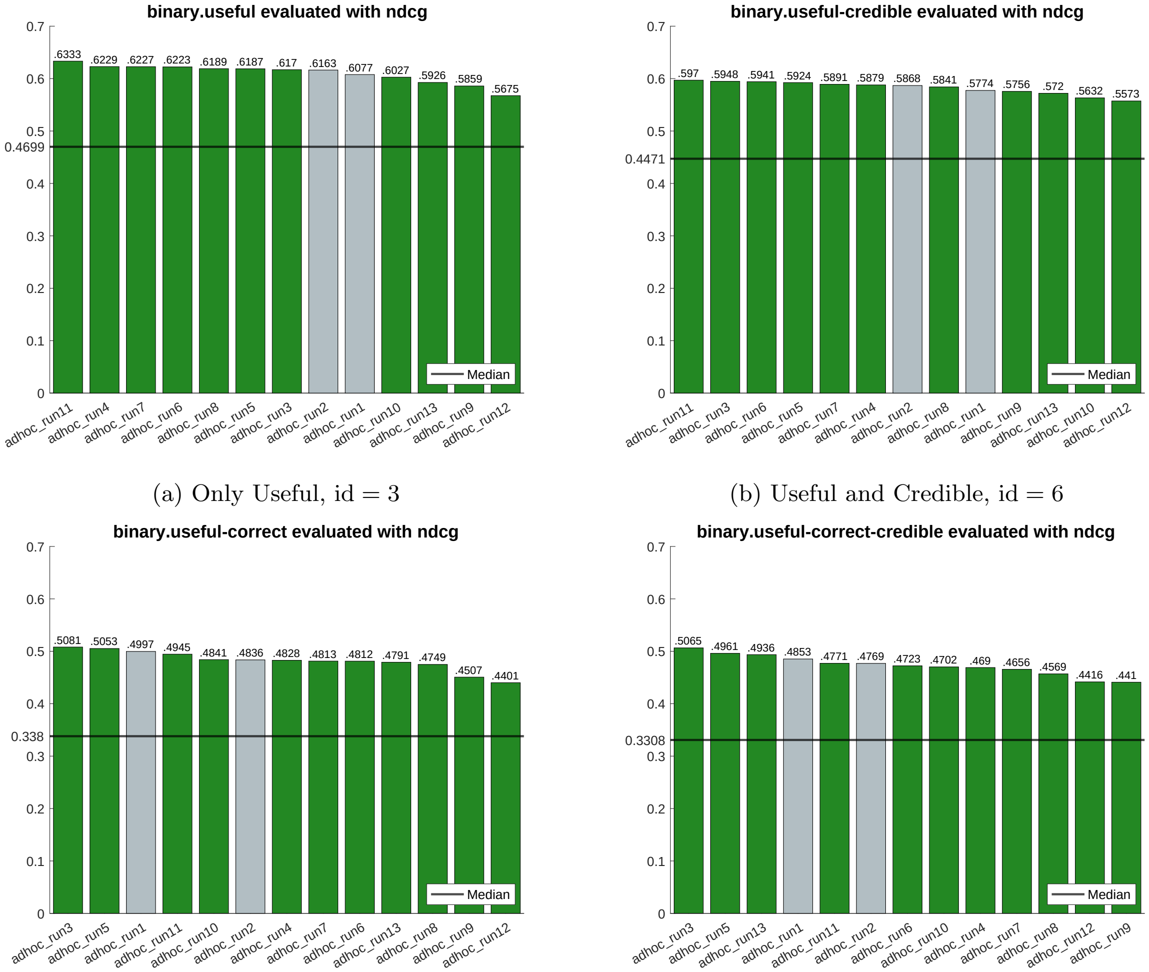
For usefulness in Figure 1a, the best run is `adhoc_run11`, which uses RRF with relevance, credibility and the reversed misinformation score. This is the best run also for usefulness and credibility in Figure 1b. Therefore, as shown by the Total Recall Task, RRF represents an effective approach to merge scores from different aspects.

Aggregating the scores with a weighted average is also quite effective: `adhoc_run3` aggregates relevance, credibility and misinformation scores with uniform weights. This is the best run with respect to usefulness and correctness in Figure 1c, usefulness, correctness and credibility in Figure 1d, and it is the second run with respect to usefulness and credibility in Figure 1b. Similarly, `adhoc_run4`, which is computed as `adhoc_run3` with RM3 instead of BM25, is the second best run with respect to usefulness in Figure 1a.

`Adhoc_run5` is the second best run for usefulness-correctness, and usefulness-correctness-credibility, in

Table 5: Auxiliary Table with mapping of qrels and measures used in Table 6.

MappingID	Qrels	Measure
0	2aspects.correct-credible	cam_map
1	2aspects.useful-credible	cam_map
2	3aspects	cam_map_three
3	binary.useful	ndcg
4	binary.useful-correct	ndcg
5	binary.useful-correct-credible	ndcg
6	binary.useful-credible	ndcg
7	graded.harmful-only	compatibility
8	graded.helpful-only	compatibility



(a) Only Useful, id = 3

(b) Useful and Credible, id = 6

(c) Useful and Correct, id = 4

(d) Useful, Correct and Credible, id = 5

Figure 1: Ad Hoc Task: Our runs evaluated with nDCG by mapping multi-aspect labels to binary labels.

Table 6: Ad Hoc Task: our runs compared to the overall median results of submitted runs. For 7 the lower the score the better.

RUNID	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
adhoc_run1	0.1911	0.2765	0.2369	0.6077	0.4997	0.4853	0.5774	0.1197	0.3679
adhoc_run10	0.2084	0.3088	0.2631	0.6027	0.4841	0.4702	0.5632	0.1263	0.3736
adhoc_run11	0.2105	0.3354	0.2823	**0.6333**	0.4945	0.4771	**0.5970**	0.1456	0.3985
adhoc_run12	0.1558	0.2563	0.2124	0.5675	0.4401	0.4416	0.5573	**0.0826**	0.3391
adhoc_run13	0.2103	0.2935	0.2491	0.5926	0.4791	0.4936	0.5720	0.1092	0.3869
adhoc_run2	**0.2155**	0.3318	0.2797	0.6163	0.4836	0.4769	0.5868	0.1446	0.4074
adhoc_run3	0.2070	0.2924	0.2495	0.6170	**0.5081**	**0.5065**	0.5948	0.1208	0.4012
adhoc_run4	0.2097	0.3302	0.2781	0.6229	0.4828	0.4690	0.5879	0.1385	0.4098
adhoc_run5	0.1994	0.2919	0.2488	0.6187	0.5053	0.4961	0.5924	0.1247	0.3981
adhoc_run6	0.2128	**0.3389**	**0.2843**	0.6223	0.4812	0.4723	0.5941	0.1487	**0.4143**
adhoc_run7	0.2086	0.3297	0.2777	0.6227	0.4813	0.4656	0.5891	0.1355	0.4023
adhoc_run8	0.2012	0.3241	0.2722	0.6189	0.4749	0.4569	0.5841	0.1338	0.3846
adhoc_run9	0.1761	0.3008	0.2491	0.5859	0.4507	0.4410	0.5756	0.0893	0.3473
median (TREC)	0.1003	0.1717	0.1389	0.4699	0.338	0.3308	0.4471	0.0747	0.3337

Figure 1c and Figure 1d respectively. This run also exploits a weighted average with BM25, where more weight is assigned to relevance (0.6) and less to credibility and reversed misinformation (0.2).

On the other side, all the worst runs exploit a single aspect, instead of applying a merging strategy. Adhoc_run12 is the worst run with respect to usefulness in Figure 1a, usefulness and credibility in Figure 1b, usefulness and correctness in Figure 1c, and is the second-to-last run with respect to all aspects in Figure 1d. Similarly, adhoc_run9 is the worst or the second-to-last run in Figures 1d, 1a and 1c. Both these runs simply re-rank the top 200 and top 100 documents just with the credibility score. Something analogous happens with adhoc_run13, which re-ranks documents just with the reversed misinformation score. This is among the worst runs for usefulness, usefulness-credibility, usefulness-correctness in Figures 1a, 1b and 1c. This results suggest that if we want to optimize a run for relevance, credibility and correctness we need to use all 3 scores.

Figure 2 shows ours runs evaluated with CAM computed with AP. Observe that the main difference between Figure 1 and Figure 2 is that in the former case document labels are aggregated across aspects and then mapped to a single binary label, while in the latter case each aspect is treated separately with AP and then CAM computes the average across aspects.

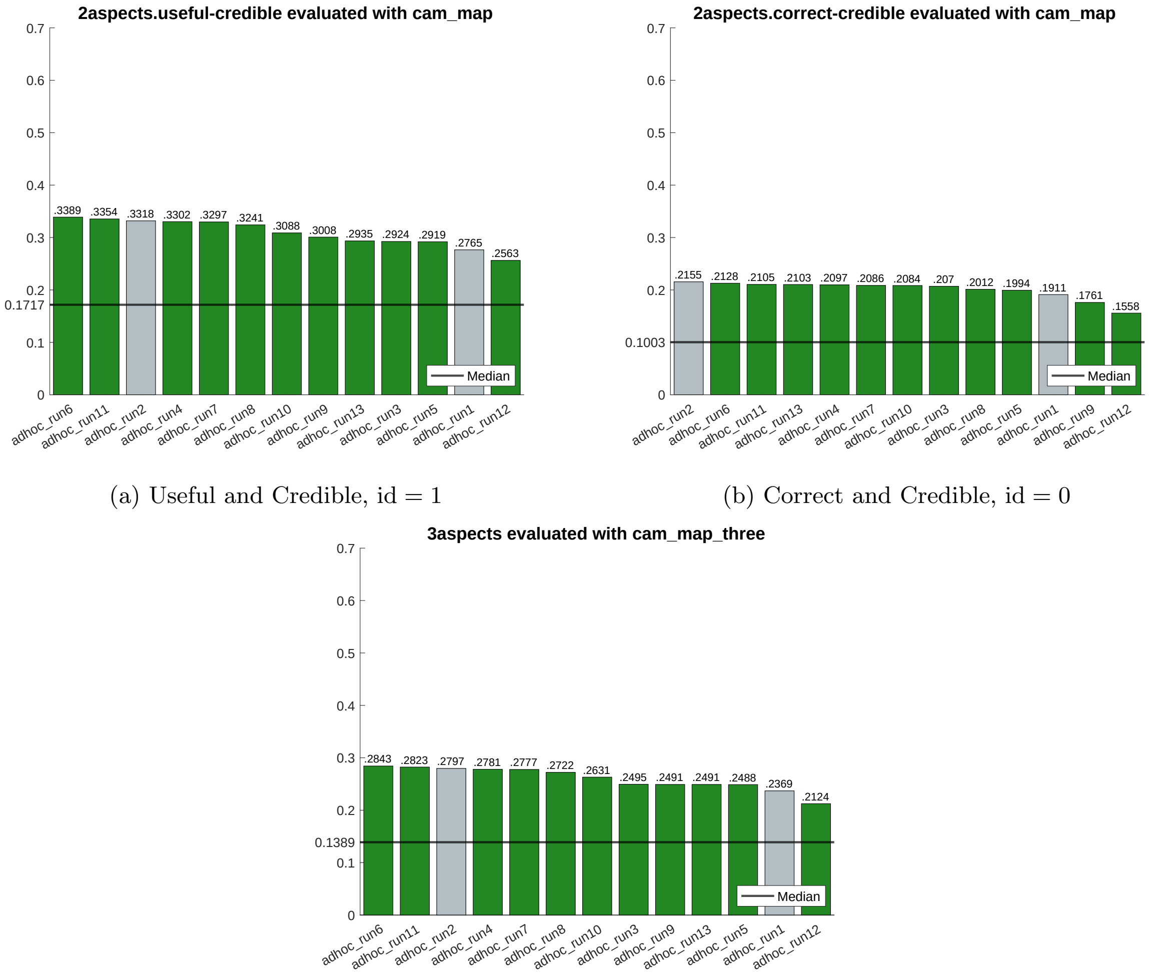
The best 3 runs within all the sub-figures of Figure 2 are: adhoc_run6, adhoc_run11 and adhoc_run2. Adhoc_run2 is one of our internal baseline, obtained with RM3 and without re-ranking. This might be due to the usefulness AP score being much higher that the credibility and correctness AP scores, thus leading CAM to reward more runs which perform quite well in terms of usefulness.

A similar reason can explain why adhoc_run6 is the best run with respect to usefulness-credibility in Figure 2a, usefulness-credibility-correctness in Figure 2c, and the second best run with respect to correctness and credibility in Figure 2b.

Finally, as observed for binary evaluation and the Total Recall Task, the RRF exploited by adhoc_run11 is an effective way to merge the runs obtained with relevance, credibility and reversed misinformation. Indeed, adhoc_run11 is the second best run for with respect to usefulness-credibility and usefulness-credibility-correctness in Figures 2c and 2a, and the third best run with respect to correctness and credibility in Figure 2b.

Figure 3 shows harmfulness and helpfulness computed with compatibility. Recall that for harmfulness the lower the score the better the run, while for helpfulness the higher the score the better the run. As clearly illustrated in Figure 3a, our runs are all worse than the task median for harmfulness, while they are all above the task median for helpfulness in Figure 3b.

Figure 4 plots harmfulness against helpfulness. The goal is to minimize harmfulness while maximizing helpfulness at the same time. Therefore, the closer a run is to the top left corner, the better the performance.



(a) Useful and Credible, id = 1

(b) Correct and Credible, id = 0

(c) Useful, Correct and Credible, id = 2

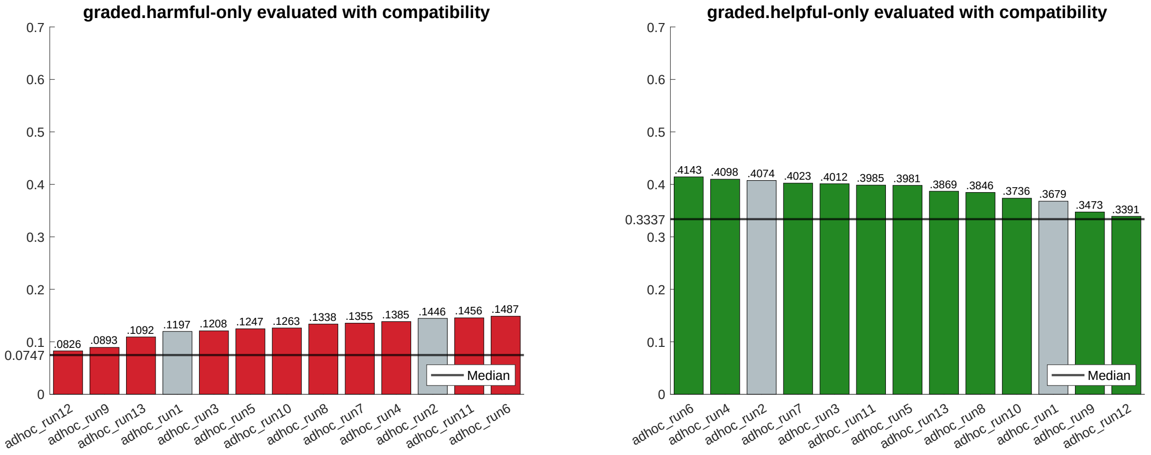
Figure 2: Ad Hoc Task: Our runs evaluated with CAM.

From Figure 4, `adhoc_run13` is the closest run to the top left corner. `Adhoc_run13` re-ranks the top 200 documents just with the reversed misinformation score. As observed with the results of the Total Recall task, the misinformation score can be used to identify harmful documents and move them towards the end of the ranking.

`adhoc_run13` and `adhoc_run10` are basically the same run, but `adhoc_run13` re-ranks the top 200 documents, while `adhoc_run10` re-ranks the top 100. This comparison suggests that if we consider higher cut-offs we might be able to reduce the harmfulness score even more.

In the bottom right corner of Figure 4 there are `adhoc_run12` and `adhoc_run9`. These two runs are obtained by re-ranking the top 200 and 100 documents just with the credibility score. These are our best runs in terms of harmfulness. Again, by increasing the re-ranking cut-off, we decrease harmfulness.

A promising strategy to optimize harmfulness and helpfulness simultaneously might be to find a better way to aggregate the misinformation and credibility score. Indeed, from Figure 4 all runs which aggregate the 3 scores increase helpfulness, but are not able to decrease harmfulness.



(a) Harmful, id = 7

(b) Helpful, id = 8

Figure 3: Ad Hoc Task: Our runs evaluated with Compatibility.

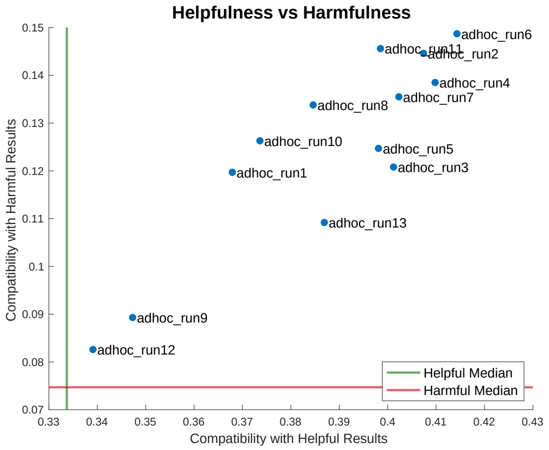


Figure 4: Ad Hoc Task: Harmful scores against helpful scores. For harmful scores the lower the better.

5 Related Work

5.1 Automatic Classification of Credibility

Olteanu et al. [24] present a list of features used to train different machine learning models able to predict the credibility of Web pages. They divide those features in 2 categories, features based on the content, as text readability, number of ads in the Web page, etc, and features based on social interactions, as number of shares on Facebook/Twitter, number of URLs, PageRank score, etc.

Similarly, Kakol et al. [15] present an automatic approach to classify the credibility of Web pages. The authors identify different features to classify credibility, as reading easiness, inclusion of links and references, language quality, the presence of author information, etc.

5.2 Automatic Fact Checking

Automatic fact checking is typically either approached using propagation-based approaches, which determine the veracity of claims based on the spread of those claims in networks [32, 20, 28, 14], or, as in this paper, content-based ones [31, 7, 23].

Content-based approaches consist of multiple tasks: claim check-worthiness detection to identify relevant claims to fact-check [2, 13, 35]; evidence document retrieval or reranking, if candidate documents are provided [37, 1]; stance detection to determine whether retrieved documents agree or disagree with the claim or neither of those [5, 8, 25, 12]; and finally, veracity prediction [31, 7, 3].

For stance detection, which is used to rank misinformation in this work, commonly studied research challenges include the prediction of stances towards unseen targets [5, 36], involving multiple targets or evidence documents at the same time [30, 22, 33], the unification of different stance label schemes [6] and transfer learning [26]. State-of-the-art architectures for stance detection typically include large pre-trained Transformer models, as we also utilize here.

6 Conclusions

In this paper, we describe our participation in the TREC Health Misinformation Track 2020. We submitted 11 runs to the Total Recall task and 13 runs to the Ad Hoc task. Our approach consists of 3 steps: (1) we create an initial run with BM25 or RM3; (2) we estimate credibility and misinformation scores for each document in the initial run; (3) we merged the scores across relevance, credibility and misinformation to re-rank documents.

Experimental results show that all our runs are above the task median, except for harmfulness. However, when it comes to the estimation of credibility and misinformation, our approach seems promising. Therefore, to minimize harmfulness, we need to find a better way to identify harmful documents by aggregating misinformation and credibility scores. Indeed, as shown in the experimental results, approaches as the weighted average or RRF tend to optimize helpfulness at the expense of harmfulness.

References

[1] L. Allein, I. Augenstein, and M.-F. Moens. Time-Aware Evidence Ranking for Fact-Checking. *arXiv preprint arXiv:2009.06402*, 2020.

[2] P. Atanasova, A. Barron-Cedeno, T. Elsayed, R. Suwaileh, W. Zaghouani, S. Kyuchukov, G. D. S. Martino, and P. Nakov. Overview of the CLEF-2018 CheckThat! Lab on Automatic Identification and Verification of Political Claims. Task 1: Check-Worthiness. *arXiv preprint arXiv:1808.05542*, 2018.

[3] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.656. URL `https://www.aclweb.org/anthology/2020.acl-main.656`.

[4] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva. Stance Detection with Bidirectional Conditional Encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, 2016.

[5] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva. Stance Detection with Bidirectional Conditional Encoding. In J. Su, X. Carreras, and K. Duh, editors, *EMNLP*, pages 876–885. The Association for Computational Linguistics, 2016. ISBN 978-1-945626-25-8. URL `http://dblp.uni-trier.de/db/conf/emnlp/emnlp2016.html#AugensteinRVB16`.

[6] I. Augenstein, S. Ruder, and A. Søgaard. Multi-task learning of pairwise sequence classification tasks over disparate label spaces. In M. A. Walker, H. Ji, and A. Stent, editors, *NAACL-HLT*, pages 1896–1906. Association for Computational Linguistics, 2018. ISBN 978-1-948087-27-8. URL `http://dblp.uni-trier.de/db/conf/naacl/naacl2018-1.html#AugensteinRS18`.

[7] I. Augenstein, C. Lioma, D. Wang, L. C. Lima, C. Hansen, C. Hansen, and J. G. Simonsen. 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, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1475.

[8] R. Baly, M. Mohtarami, J. R. Glass, L. Màrquez, A. Moschitti, and P. Nakov. Integrating Stance Detection and Fact Checking in a Unified Corpus. In M. A. Walker, H. Ji, and A. Stent, editors, *NAACL-HLT (2)*, pages 21–27. Association for Computational Linguistics, 2018. ISBN 978-1-948087-29-2. URL `http://dblp.uni-trier.de/db/conf/naacl/naacl2018-2.html#BalyMGMMN18`.

[9] C. Clarke, M. Maistro, M. Smucker, and G. Zuccon. Overview of the TREC 2020 Health Misinformation Track (to appear). In E. M. Voorhees and A. Ellis, editors, *Proceedings of the Twenty-Nine Text REtrieval Conference, TREC 2020, Gaithersburg, Maryland, USA, November 16-19, 2020*, volume - of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020.

[10] G. Cormack, C. Clarke, and S. Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. pages 758–759, 01 2009. doi: 10.1145/1571941.1572114.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, 2019.

[12] W. Ferreira and A. Vlachos. Emergent: a novel data-set for stance classification. In K. Knight, A. Nenkova, and O. Rambow, editors, *HLT-NAACL*, pages 1163–1168. The Association for Computational Linguistics, 2016. ISBN 978-1-941643-91-4. URL `http://dblp.uni-trier.de/db/conf/naacl/naacl2016.html#FerreiraV16`.

[13] C. Hansen, C. Hansen, S. Alstrup, J. Grue Simonsen, and C. Lioma. Neural Check-Worthiness Ranking With Weak Supervision: Finding Sentences for Fact-Checking. In *Companion Proceedings of the 2019 World Wide Web Conference*, pages 994–1000, 2019.

[14] M. Hartmann, Y. Golovchenko, and I. Augenstein. Mapping (dis-)information flow about the mh17 plane crash. *CoRR*, abs/1910.01363, 2019. URL `http://dblp.uni-trier.de/db/journals/corr/corr1910.html#abs-1910-01363`.

[15] M. Kakol, R. Nielek, and A. Wierzbicki. Understanding and Predicting Web Content Credibility Using the Content Credibility Corpus. *Information Processing & Management*, 53(5):1043–1061, 2017.

[16] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 120–127, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581133316.

[17] C. Lioma, B. Larsen, W. Lu, and Y. Huang. A study of factuality, objectivity and relevance: Three desiderata in large-scale information retrieval? In *Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*, BDCAT '16, page 107–117, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450346177. doi: 10.1145/3006299.3006315. URL `https://doi.org/10.1145/3006299.3006315`.

[18] C. Lioma, J. G. Simonsen, and B. Larsen. Evaluation measures for relevance and credibility in ranked lists. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '17, page 91–98, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450344906. doi: 10.1145/3121050.3121072. URL `https://doi.org/10.1145/3121050.3121072`.

[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[20] Z. Liu, C. Xiong, M. Sun, and Z. Liu. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.655.

[21] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry. SemEval-2016 Task 6: Detecting Stance in Tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, 2016.

[22] M. Mohtarami, R. Baly, J. Glass, P. Nakov, L. Màrquez, and A. Moschitti. Automatic Stance Detection Using End-to-End Memory Networks. 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 767–776, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1070. URL `https://www.aclweb.org/anthology/N18-1070`.

[23] Y. Nie, H. Chen, and M. Bansal. Combining Fact Extraction and Verification with Neural Semantic Matching Networks. In *AAAI*, pages 6859–6866. AAAI Press, 2019. ISBN 978-1-57735-809-1. URL `http://dblp.uni-trier.de/db/conf/aaai/aaai2019.html#NieCB19`.

[24] A. Olteanu, S. Peshterliev, X. Liu, and K. Aberer. Web credibility: Features exploration and credibility prediction. In P. Serdyukov, P. Braslavski, S. O. Kuznetsov, J. Kamps, S. Rüger, E. Agichtein, I. Segalovich, and E. Yilmaz, editors, *Advances in Information Retrieval*, pages 557–568, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-36973-5.

[25] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel. A simple but tough-to-beat baseline for the Fake News Challenge stance detection task. *CoRR*, abs/1707.03264, 2017. URL `http://dblp.uni-trier.de/db/journals/corr/corr1707.html#RiedelASR17`.

[26] B. Schiller, J. Daxenberger, and I. Gurevych. Stance Detection Benchmark: How Robust Is Your Stance Detection? *CoRR*, abs/2001.01565, 2020. URL `http://dblp.uni-trier.de/db/journals/corr/corr2001.html#abs-2001-01565`.

[27] J. Schwarz and M. R. Morris. Augmenting web pages and search results to support credibility assessment. In *ACM Conference on Computer-Human Interaction*, May 2011. URL https://www.microsoft.com/en-us/research/publication/augmenting-web-pages-and-search-results-to-support-credibility-assessment/.

[28] K. Shu, S. Wang, and H. Liu. Beyond news contents: The role of social context for fake news detection. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, pages 312–320, New York, 2019. Association for Computing Machinery. doi: 10.1145/3289600.3290994.

[29] V. Slovikovskaya and G. Attardi. Transfer Learning from Transformers to Fake News Challenge Stance Detection (FNC-1) Task. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1211–1218, 2020.

[30] P. Sobhani, D. Inkpen, and X. Zhu. A Dataset for Multi-Target Stance Detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 551–557. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/E17-2088.

[31] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, 2018.

[32] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 647–653, Vancouver, 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2102.

[33] Z. Wang, Q. Wang, C. Lv, X. Cao, and G. Fu. Unseen Target Stance Detection with Adversarial Domain Generalization. In *IJCNN*, pages 1–8. IEEE, 2020. ISBN 978-1-7281-6926-2. URL http://dblp.uni-trier.de/db/conf/ijcnn/ijcnn2020.html#WangWLCF20.

[34] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, abs/1910.03771, 2019.

[35] D. Wright and I. Augenstein. Claim Check-Worthiness Detection as Positive Unlabelled Learning. In *Findings of EMNLP*. Association for Computational Linguistics, 2020.

[36] B. Xu, M. Mohtarami, and J. R. Glass. Adversarial Domain Adaptation for Stance Detection. *CoRR*, abs/1902.02401, 2019. URL http://dblp.uni-trier.de/db/journals/corr/corr1902.html#abs-1902-02401.

[37] W. Yin and D. Roth. TwoWingOS: A Two-Wing Optimization Strategy for Evidential Claim Verification. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *EMNLP*, pages 105–114. Association for Computational Linguistics, 2018. ISBN 978-1-948087-84-1. URL http://dblp.uni-trier.de/db/conf/emnlp/emnlp2018.html#0001R18.