A Context-Aware Approach for Detecting Worth-Checking Claims in Political Debates

Pepa Gencheva[1], Preslav Nakov[2], Lluís Màrquez[2], Alberto Barrón-Cedeño[2], and Ivan Koychev[1]

[1]Sofia University "St. Kliment Ohridski", Bulgaria
[2]Qatar Computing Research Institute, HBKU, Qatar
pepa.k.gencheva@gmail.com, {pnakov, lmarquez, albarron}@hbku.edu.qa
koychev@fmi.uni-sofia.bg

Abstract

In the context of investigative journalism, we address the problem of automatically identifying which claims in a given document are most worthy and should be prioritized for fact-checking. Despite its importance, this is a relatively understudied problem. Thus, we create a new corpus of political debates, containing statements that have been fact-checked by nine reputable sources, and we train machine learning models to predict which claims should be prioritized for fact-checking, i.e., we model the problem as a ranking task. Unlike previous work, which has looked primarily at sentences in isolation, in this paper we focus on a rich input representation modeling the context: relationship between the target statement and the larger context of the debate, interaction between the opponents, and reaction by the moderator and by the public. Our experiments show state-of-the-art results, outperforming a strong rivaling system by a margin, while also confirming the importance of the contextual information.

1 Introduction

The current coverage of the political landscape in the press and in social media has led to an unprecedented situation. Like never before, a statement in an interview, a press release, a blog note, or a tweet can spread almost instantaneously and reach the public in no time. This proliferation speed has left little time for double-checking claims against the facts, which has proven critical in politics, e.g., during the 2016 presidential campaign in the USA, which was arguably impacted by fake news in social media and by false claims.

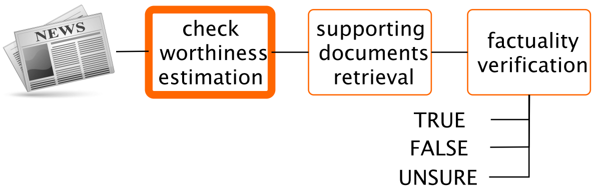

Figure 1: Information verification pipeline.

Investigative journalists and volunteers have been working hard trying to get to the root of a claim and to present solid evidence in favor or against it. Manual fact-checking has proven very time-consuming, and thus automatic methods have been proposed as a way to speed-up the process. For instance, there has been work on checking the factuality/credibility of a claim, of a news article, or of an information source (Castillo et al., 2011; Ba et al., 2016; Zubiaga et al., 2016; Ma et al., 2016; Hardalov et al., 2016; Karadzhov et al., 2017a,b; Nakov et al., 2017). However, less attention has been paid to other steps of the fact-checking pipeline, which is shown in Figure 1.

The process starts when a document is made public. First, an intrinsic analysis is carried out in which check-worthy text fragments are identified. Then, other documents that might support or rebut a claim in the document are retrieved from various sources. Finally, by comparing a claim against the retrieved evidence, a system can determine whether the claim is likely true or likely false. For instance, Ciampaglia et al. (2015) do this on the basis of a knowledge graph derived from Wikipedia. The outcome could then be presented to a human expert for final judgment.[1]

[1] As of present, fully automatic methods for fact checking still lag behind in terms of quality, and thus also of credibility in the eyes of the users, compared to what high-quality manual checking by reputable sources can achieve, which means that a final double-checking by a human expert is needed.

Proceedings of Recent Advances in Natural Language Processing, pages 267–276,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_037

In this paper, we focus on the first step: predicting check-worthiness of claims. Our contributions can be summarized as follows:

1. *New dataset:* We build a new dataset of manually-annotated claims, extracted from the 2016 US presidential and vice-presidential debates, which we gathered from nine reputable sources such as CNN, NPR, and PolitiFact, and which we release to the research community.

2. *Modeling the context:* We develop a novel approach for automatically predicting which claims should be prioritized for fact-checking, based on a rich input representation. In particular, we model not only the textual content, but also the context: how the target claim relates to the current segment, to neighboring segments and sentences, and to the debate as a whole, and also how the opponents and the public react to it.

3. State-of-the-art results: We achieve state-of-the-art results, outperforming a strong rivaling system by a margin, while also demonstrating that this improvement is due primarily to our modeling of the context.

We model the problem as a ranking task, and we train both Support Vector Machines (SVM) and Feed-forward Neural Networks (FNN) obtaining state-of-the-art results. We also analyze the relevance of the specific feature groups and we show that modeling the context yields a significant boost in performance. Finally, we also analyze whether we can learn to predict which facts are worth-checking with respect to each of the individual media sources, thus capturing their biases. It is worth noting that while trained on political debates, many features of our model can be potentially applied to other kinds of information sources, e.g., interviews and news.

The rest of the paper is organized as follows: Section 2 overviews related work. Section 3 describes the process of gathering and annotating the debates dataset. Section 4 describes our supervised approach to predicting fact-checking worthiness, including the explanation of the model and the information sources we use. Section 5 includes the evaluation and discusses the results. Section 6 provides further analysis. Finally, Section 7 presents the conclusions and outlines some lines for future research.

2 Related Work

The previous work that is most relevant to our work here is that of (Hassan et al., 2015), who developed the *ClaimBuster* system, which assigns each sentence in a document a score, i.e., a number between 0 and 1 showing how worthy it is for fact-checking. The system is trained on their own dataset of about 8 thousand debate sentences (1,673 of them worth-checking), annotated by students, university professors, and journalists. Unfortunately, this dataset is not publicly available, and contains sentences without context as about 60% of the original sentences had to be thrown away due to lack of agreement.

In contrast, we develop a new publicly-available dataset,[2] based on manual annotations of political debates by nine highly-reputed fact-checking sources, where sentences are annotated in the context of the entire debate. This allows us to explore a novel approach, which focuses on the context.

Note also that the *ClaimBuster* dataset is annotated following guidelines from (Hassan et al., 2015) rather than a real fact-checking website; yet, it was evaluated against CNN and PolitiFact (Hassan et al., 2016). In contrast, we train and evaluate directly on annotations from fact-checking websites, and thus we learn to fit them better.

Beyond the document context, it has been proposed to mine check-worthy claims on the Web. For example, Ennals et al. (2010a) searched for linguistic cues of disagreement between the author of a statement and what is believed, e.g., "falsely claimed that X". The claims matching the patterns go through a statistical classifier, which marks the text of the claim. This procedure can be used to acquire a corpus of disputed claims from the Web.

Given a set of disputed claims, (Ennals et al., 2010b) approached the task as locating new claims on the Web that entail the ones that have already been collected. Thus, the task can be conformed as recognizing textual entailment, which is analyzed in detail in (Dagan et al., 2009).

Finally, Le et al. (2016) argued that the top terms in claim vs. non-claim sentences are highly overlapping, which is a problem for bag-of-words approaches. Thus, they used a Convolutional Neural Network, where each word is represented by its embedding and each named entity is replaced by its tag, e.g., *person*, *organization*, *location*.

[2] The dataset and the source code are available in GitHub: `https://github.com/pgencheva/claim-rank`

Medium	1st	2nd	VP	3rd	Total
ABC News	35	50	29	28	142
Chicago Tribune	30	29	31	38	128
CNN	46	30	37	60	173
FactCheck.org	15	45	47	60	167
NPR	99	92	91	89	371
PolitiFact	74	62	60	57	253
The Guardian	27	39	54	72	192
The New York Times	26	25	46	52	149
The Washington Post	26	19	33	17	95
Total annotations	378	391	428	473	1,670
Annotated sentences	218	235	183	244	880

Table 1: Number of annotations in each medium for the 1st, 2nd and 3rd presidential and the vice-presidential debates.

Agreement Level	Number of Sentences	Cumulative Sum
9	1	1
8	6	7
7	5	12
6	19	31
5	26	57
4	40	97
3	100	197
2	191	388
1	**492**	**880**
Total number of sentences: 5,415		

Table 2: Agreement between the media represented as the number of sentences that n out of nine providers identified as worth-checking.

3 The CW-USPD-2016 Corpus on US Presidential Debates

We created a new dataset called CW-USPD-2016 (check-worthiness in the US presidential debates 2016) for finding check-worthy claims in context. In particular, we used four transcripts of the 2016 US election: one vice-presidential and three presidential debates. For each debate, we used the publicly-available manual analysis about it from nine reputable fact-checking sources, as shown in Table 1. This could include not just a statement about factuality, but any free text that journalists decided to add, e.g., links to biographies or behavioral analysis of the opponents and moderators. We converted this to binary annotation about whether a particular sentence was annotated for factuality by a given source. Whenever one or more annotations were about part of a sentence, we selected the entire sentence, and when an annotation spanned over multiple sentences, we selected each of them.

Ultimately, we ended up with a corpus of four debates, with a total of 5,415 sentences. The agreement between the sources was low as Table 2 shows: only one sentence was selected by all nine sources, 57 sentences by at least five, 197 by at least three, 388 by at least two, and 880 by at least one. The reason for this is that the different media aimed at annotating sentences according to their own editorial line, rather than trying to be exhaustive in any way. This suggests that the task of predicting which sentence would contain worth-checking claims will be challenging. Thus, below we focus on a ranking task rather than on absolute predictions. Moreover, we predict which sentence would be selected (*i*) by at least one of the media, or (*ii*) by a specific medium.

Note that the investigative journalists did not select the check-worthy claims in isolation. Our analysis shows that these include claims that were highly disputed during the debate, that were relevant to the topic introduced by the moderator, etc. We will make use of these contextual dependencies below, which is something that was not previously tried in related work.

4 Modeling Check-Worthiness

We developed a rich input representation in order to model and to learn the *check-worthiness* concept. The feature types we implemented operate at the sentence- (S) and at the context-level (C), in either case targeting *segments* by the same speaker.[3] The context features are novel and a contribution of this study. We also implemented a set of core features to compare to the state of the art. All of them are described below.

4.1 Sentence-Level Features

ClaimBuster-based (*1,045 S features*; core): First, in order to be able to compare our model and features directly to the previous state of the art, we re-implemented, to the best of our ability, the sentence-level features of *ClaimBuster* as described in (Hassan et al., 2015), namely TF-IDF-weighted bag of words (998 features), part-of-speech tags (25 features), name entities as recognized by *Alchemy API*[4] (20 features), sentiment score from Alchemy API (1 feature), and number of tokens in the target sentence (1 feature).

[3] We define a *segment* as a maximal set of consecutive sentences by the same speaker without intervention by another speaker or by the moderator.

[4] http://www.ibm.com/watson/alchemy-api.html

Apart from providing means of comparison to the state of the art, these features also make a solid contribution to the final system we build for claim-worthiness estimation. However, note that we did not have access to the training data of ClaimBuster, which is not publicly available, and we thus train on our own dataset.

Sentiment (*2 S features*): Some sentences are highly negative, which can signal the presence of an interesting claim to check, as the two example sentences below show (from the 1st and the 2nd presidential debates):

Trump:	Murders are up.
Clinton:	Bullying is up.

We used the NRC sentiment lexicon (Mohammad and Turney, 2013) as a source of words and n-grams with positive/negative sentiment, and we counted the number of positive and of negative words in the target sentence. These features are different from those in the *CB features* above, where these lexicons were not used.

Named entities (NE) (*1 S feature*): Sentences that contain named entity mentions are more likely to contain a claim that is worth fact-checking as they discuss particular people, organizations, and locations. Thus, we have a feature that counts the number of named entities in the target sentence; we use the *NLTK toolkit* for named entity recognition (Loper and Bird, 2002). Unlike the *CB features* above, here we only have one feature; we also use a different toolkit for named entity recognition.

Linguistic features (*9 S features*): We count the number of words in each sentence that belong to each of the following lexicons: Language Bias lexicon (Recasens et al., 2013), Opinion Negative and Positive Words (Liu et al., 2005), Factives and Assertive Predicates (Hooper, 1974), Hedges (Hyland, 1998), Implicatives (Karttunen, 1971), and Strong and Weak subjective words. Some examples are shown in Table 3.

Feature Name	Examples
Bias	capture, create, demand, follow
Negatives	abnormal, bankrupt, cheat, conflicts
Positives	accurate, achievements, affirm
Factives	realize, know, discover, learn
Assertives	think, believe, imagine, guarantee
Hedges	approximately, estimate, essentially
Implicatives	cause, manage, hesitate, neglect
Strong-subj	admire, afraid, agreeably, apologist
Weak-subj	abandon, adaptive, champ, consume

Table 3: Linguistic features and examples.

Tense (*1 S feature*): Most of the check-worthy claims mention past events. In order to detect when the speaker is making a reference to the past or s/he is talking about his/her future vision and plans, we include a feature with three values—indicating whether the text is in past, present of future tense. The feature is extracted from the verbal expressions, using POS tags and a list of auxiliary verbs and phrases such as *will*, *have to*, etc.

Length (*1 S feature*): Shorter sentences are generally less likely to contain a worth-checking claim.[5] Thus, we have a feature for the length of the sentence in terms of characters. Note that this feature was not part of the *CB features*, as there length was modeled in terms of tokens, but here we do so using characters.

4.2 Contextual Features

Position (*3 C features*): A sentence on the boundaries of a speaker's segment could contain a reaction to another statement or could provoke a reaction, which in turn could signal a worth-checking claim. Thus, we added information about the position of the target sentence in its segment: whether it is first/last, as well as its reciprocal rank in the list of sentences in that segment.

Segment sizes (*3 C features*): The size of the segment belonging to one speaker might indicate whether the target sentence is part of a long speech, makes a short comment or is in the middle of a discussion with lots of interruptions. The size of the previous and of the next segments is also important in modeling the dialogue flow. Thus, we include three features with the sizes of the previous, the current and the next segments.

Metadata (*8 C features*): Worth-checking claims often contain accusations about the opponents, as the example below shows (from the 2nd presidential debate):

Trump:	**Hillary Clinton** attacked those same women and attacked them viciously.
Clinton:	They're doing it to try to influence the election for **Donald Trump**.

Thus, we use a feature that indicates whether the target sentence mentions the name of the opponent, whether the speaker is the moderator, and also who is speaking (3 features). We further use three binary features, indicating whether the target sentence is followed by a system message: *applause*, *laugh*, or *cross-talk*.

[5] One notable exception are short sentences with negations, e.g., *Wrong.*, *Nonsense.*, etc.

4.3 Mixed Features

The feature groups in this subsection contain a mixture of sentence- and of contextual-level features. For example, if we use a discourse parser to parse the target sentence only, any features we extract from the parse would be sentence-level. However, if we parse an entire segment, we would also have contextual features.

Topics (*300+3 S+C features*): Some topics are more likely to be associated with worth-checking claims, and thus we have features modeling the topics in the target sentence as well as in the surrounding context. We trained a Latent Dirichlet Allocation (LDA) topic model (Blei et al., 2003) on all political speeches and debates in *The American Presidency Project*[6] using all US presidential debates in the 2007–2016 period[7]. We had 300 topics, and we used the distribution over the topics as a representation for the target sentence. We further modeled the context using cosines with such representations for the previous, the current, and the next segment.

Embeddings (*300+3 S+C features*): We further modeled semantics using word embeddings. We used the pre-trained 300-dimensional Google News word embeddings by Mikolov et al. (2013) to compute an average embedding vector for the target sentence, and we used the 300 coordinates of that vector. We also modeled the context as the cosine between that vector and the vectors for three segments: the previous, the current, and the following one.

Discourse (*2+18 S+C features*): We saw above that contradiction can signal the presence of worth-checking claims, and contradiction can be expressed by a discourse relation such as CONTRAST. As other discourse relations such as BACKGROUND, CAUSE, and ELABORATION can also be useful, we used a discourse parser (Joty et al., 2015) to parse the entire segment, and we focused on the relationship between the target sentence and the other sentences in its segment; this gave rise to 18 contextual indicator features. We further analyzed the internal structure of the target sentence —how many nuclei and how many satellites it contains—, which gave rise to two sentence-level features.

Contradictions (*1+4 S+C features*): Many claims selected for fact-checking contain contradictions to what has been said earlier, as in the example below (from the 3rd presidential debate):

Clinton:	[. . .] about a potential nuclear competition in Asia, you said, you know, go ahead, enjoy yourselves, folks.
Trump:	**I didn't say** nuclear.

We model this by counting the negations in the target sentence as found in a dictionary of negation cues such as *not*, *didn't*, and *never*. We further model the context as the number of such cues in the two neighboring sentences from the same segment and the two neighboring segments.

kNN (*2+1 S+C features*): We used three more features inspired by k-nearest neighbor (kNN) classification. The first one (sentence-level) uses the maximum over the training sentences of the number of matching words between the testing and the training sentence, which is further multiplied by -1 if the latter was not worth-checking. We also used another version of the feature, where we multiplied it by 0 if the speakers were different (contextual). A third version took as a training set all claims checked by *PolitiFact* (excluding the target sentence).

5 Experiments and Evaluation

5.1 Experimental Setting

We experimented with two learning algorithms. The first one is an SVM classifier with an RBF kernel.[8] The second one is a deep feed-forward neural network (FNN) with two hidden layers (with 200 and 50 neurons, respectively) and a softmax output unit for the binary classification. We used ReLU (Glorot et al., 2011) as the activation function and we trained the network with Stochastic Gradient Descent (LeCun et al., 1998).

The models were trained to classify sentences as positive if *one or more media* had fact-checked a claim inside the target sentence, and negative otherwise. We then used the classifier scores to rank the sentences with respect to *check-worthiness*.[9] We tuned the parameters and we evaluated the performance using 4-fold cross-validation, using each of the four debates in turn for testing while training on the remaining three ones.

[6] http://www.presidency.ucsb.edu/debates.php

[7] https://github.com/paigecm/2016-campaign

[8] The RBF kernel was clearly superior to a linear kernel in our initial experiments.

[9] We also tried using ordinal regression, and SVM-perf, an instantiation of SVM-struct, to directly optimize precision, but none of them yielded improvements.

For evaluation, we used ranking measures such as *Precision at* k ($P@k$) and *Mean Average Precision* (MAP). As Table 1 shows, most media rarely check more than 50 claims per debate. *NPR* and *PolitiFact* are notable exceptions, the former going up to 99; yet, on average there are two claims per sentence, which means that there is no need to fact-check more than 50 sentences even for them. Thus, we report $P@k$ for $k \in \{5, 10, 20, 50\}$.[10]

MAP is the mean of the Average Precision across the four debates. The average precision for a debate is computed as follows:

$$\text{AvPrec} = \frac{\sum_{k=1}^{n}(P(k) \times rel(k))}{\text{number of relevant utterances}} \qquad (1)$$

where n is the number of sentences to rank in the debate, $P(k)$ is the precision at k and $rel(k) = 1$ if the utterance at position k is worth-checking, and it is 0 otherwise.

We also measure the recall at the R-th position of returned sentences for each debate. R is the number of relevant documents for that debate and the metric is known as R-Precision (R-Pr).

5.2 Results

Table 4 shows the performance of our models when using all features described in Section 4: see the SVM$_{All}$ and the FNN$_{All}$ rows.

In order to put the numbers in perspective, we also show the results for five increasingly competitive baselines. The first one is a random baseline. It is then followed by an SVM classifier based on a bag-of-words representation with TF-IDF weights learned on the training data. Then come three versions of the *ClaimBuster* system: CB-Platform refers to the performance of *ClaimBuster* using the scores obtained from their online demo,[11] which we accessed on December 20, 2016, and SVM$_{CBfeat}$ and FNN$_{CBfeat}$ are our reimplementations of *ClaimBuster* using their features, which we then use in our SVM or FNN classifiers trained on our dataset.

We can see that, as expected, all systems perform well above the random baseline. The three versions of ClaimBuster also outperform the TD-IDF baseline on most measures. Moreover, our reimplementations of *ClaimBuster* are better than the online platform in terms of MAP. This is expected as their system is trained on a different dataset and it may suffer from testing on slightly out-of-domain data. At the same time, this is reassuring for our implementation of the features, and allows for a more realistic comparison to the *ClaimBuster* system.

More importantly, we can see that both the SVM and the FNN versions of our system, when trained with all features, consistently outperform all three versions of *ClaimBuster* on all measures. This means that the extra information coded in our model, mainly more linguistic, structural, and contextual features, has an important contribution to the final performance.

We can further see that the neural network model, FNN$_{All}$, clearly outperforms the SVM model for this task: consistently on all metrics. As an example, with the precision values achieved by FNN$_{All}$, the system would rank on average 4 positive examples in the list of its top-5 choices, and also 14-15 in the top-20 list. Considering the recall at the first R sentences, we will be able to encounter 43% of the total number of check-worthy sentences. This is quite remarkable given the difficulty of the task.

As a next step of the evaluation, we perform error analysis of the decisions made by the Neural Network that uses all available features. We present examples of False Positives (FP) and False Negatives (FN):

1 FP Clinton: He actually was sued twice by the Justice Department.
2 FP Clinton: Five million people lost their homes.
3 FP Clinton: There's no doubt now that Russia has used cyber attacks against all kinds of organizations in our country, and I am deeply concerned about this.
4 FP Trump: Your husband signed NAFTA, which was one of the worst things that ever happened to the manufacturing industry.
5 FN Trump: This is one of the worst deals ever made by any country in history.
6 FN Trump: Well, nobody was pressing it, nobody was caring much about it.
7 FN Trump: So Ford is leaving.
8 FN Trump: It was taken away from her.

Regarding the false positive examples, we can conclude that they could be also interesting for fact-checking, as they make some questionable statements. The list of false negatives contains sentences which belong to a whole group of annotations and some of them are not check-worthy on their own such as the eighth example. Some of the

[10] Note that as far as the difference between the P@k metrics (especially between 5 and 10) is in terms of a few sentences, the deviation between them can seem large, while caused by a few correctly/wrongly predicted sentences.

[11] http://idir-server2.uta.edu/claimbuster/demo

System	MAP	R-Pr	P@5	P@10	P@20	P@50
Baselines						
Random	.164	.007	.200	.125	.138	.160
TF-IDF	.314	.333	.550	.475	.413	.360
CB Platform	.317	.349	.500	.550	.488	.405
SVM$_{CBfeat}$	.360	.393	.400	.425	.525	.495
FNN$_{CBfeat}$	.357	.379	.500	.550	.550	.510
Systems (using all features)						
SVM$_{All}$	.395	.406	.650	**.725**	.588	.565
FNN$_{All}$	**.427**	**.432**	**.800**	**.725**	**.713**	**.600**

Table 4: Evaluation results: our full systems (SVM and FNN) vs. a number of baselines:random and a TF-IDF baselines, also *ClaimBuster* from the platform, and our two reimplementations thereof.

S or C	Feat. Group	MAP	R-Pr	P@5	P@10	P@20	P@50
S+C	Embeddings	.357	.380	.450	.525	.488	.495
S+C	kNN	.313	.322	.800	.725	.612	.445
S	Linguistic	.308	.333	.450	.450	.463	.430
S	Sentiment	.260	.277	.550	.400	.288	.315
C	Metadata	.256	.268	.350	.300	.388	.370
S	Length	.254	.350	.350	.375	.400	.340
S	NEs	.236	.251	.250	.275	.313	.280
S+C	Contradiction	.222	.222	.400	.275	.288	.260
C	Segment size	.217	.231	.100	.150	.150	.245
C	Position	.212	.230	.100	.075	.175	.230
S+C	Discourse	.205	.206	.200	.300	.325	.255
S+C	Topics	.180	.178	.000	.000	.013	.085

Table 5: Performance of each feature group in isolation, using the FNN system. Results sorted by decreasing MAP score.

false negatives, though, need to be fact-checked and our model missed them such as sixth and seventh examples. An interesting observation is that we have two sentences, making the same statements using different wording - fourth and fifth sentences. On the one hand, the annotators should have labeled both of the sentences in the same manner, and on th other hand, our model should have also labeled them equally.

Finally, we can conclude that the false positives of our ranking system also make good candidates for credibility verification and demonstrate that the system has successfully extracted common patterns for check-worthiness. This way, the top-n list will contain mostly sentences which need to be further checked. Given the discrepancies and the disagreement between the annotations, a further cleaning of the corpus might be needed to prevent missing important check-worthy statements.

6 Discussion

In this section, we present some in-depth analysis and further discussion.

6.1 Individual Feature Types

Table 5 shows the performance of the individual feature types described in Section 4, when training using our FNN model, and ordered by their decreasing MAP score. We can see that *embeddings* perform best (MAP=.357, P@50=.495), which shows that modeling semantics and the similarity of a sentence against its context is quite important. Then comes *kNN* with MAP of .313 and P@50 of .455. The high performance of this feature reveals the frequent usage of statements which resemble already fact-checked ones. In the case of false claims, this can be considered as a testimony for the existence of a post-truth era (Davies, 2016).

Then follow two sentence-level features, *linguistic features* and *sentiment*, with MAP of .308 and .260, and P@50 of .430 and .315. This is on par with previous work, which has focused primarily on similar sentence-level features. Then follow a contextual feature: *Metadata* (MAP=.256, P@50=.370). And two sentence features: *length* and *named entities*, with MAP of .254 and .236, and P@50 of .340 and .280.

At the bottom of the table we find *position*, a general contextual feature with MAP of .212 and P@50 of .230, followed by *discourse* and *topics*.

6.2 Effect of Context Modeling

Next, we study the impact of the contextual features.

Table 6 shows the results when using all features vs. excluding the contextual features vs. using the contextual features only. We can see that the contextual features have a major impact on performance: excluding them yields major drop for all measures, e.g., MAP drops from .427 to .385, and P@5 drops from .800 to .550. The last two rows in the table show that using contextual features only performs about the same as *CB Platform* (which uses no contextual features at all).

System	MAP	R-Pr	P@5	P@10	P@20	P@50
All	.427	.432	.800	.725	.713	.600
All, no contextual	.385	.390	.550	.500	.550	.540
Only contextual	.317	.404	.725	.563	.465	.465
CB Platform	*.317*	*.349*	*.500*	*.550*	*.488*	*.405*

Table 6: Impact of the contextual features on the overall performance (FNN system).

System	MAP	R-Pr	P@5	P@10	P@20	P@50
PolitiFact (PF)						
CB Platform	.154	.213	.200	.300	.238	.210
NN (train on PF)	.218	.274	.450	.325	.300	.270
NN (train on all)	.213	.246	.400	.350	.375	.290
NPR						
CB Platform	.144	.186	.200	.225	.225	.180
NN (train on NPR)	.193	.216	.550	.475	.350	.255
NN (train on all)	.208	.250	.500	.450	.375	.255
The New York Times (NYT)						
CB Platform	.103	.250	.250	.163	.135	
NN (train on NYT)	.136	.178	.250	.225	.188	.135
NN (train on all)	.136	.169	.150	.200	.163	.160
The Guardian (TG)						
CB Platform	.084	.128	.100	.100	.125	.140
NN (train on TG)	.121	.156	.250	.225	.200	.155
NN (train on all)	.128	.185	.100	.150	.188	.165
FactCheck (FC)						
CB Platform	.081	.213	.150	.125	.100	.115
NN (train on FC)	.081	.098	.050	.125	.088	.085
NN (train on all)	.115	.149	.100	.125	.125	.140
CNN						
CB Platform	.082	.096	.150	.125	.088	.085
NN (train on CNN)	.079	.076	.100	.100	.100	.090
NN (train on all)	.095	.087	.000	.075	.088	.100
Chicago Tribune (CT)						
CB Platform	.053	.032	.050	.050	.038	.065
NN (train on CT)	.087	.118	.150	.150	.175	.105
NN (train on all)	.092	.098	.150	.075	.100	.090
ABC						
CB Platform	.065	.066	.150	.125	.088	.080
NN (train on ABC)	.059	.068	.050	.050	.100	.060
NN (train on all)	.088	.090	.150	.150	.113	.100
Washington Post (WP)						
CB Platform	.048	.056	.050	.075	.050	.045
NN (train on WP)	.102	.098	.200	.175	.113	.080
NN (train on all)	.076	.751	.200	.100	.075	.080

Table 7: Training on the target medium vs. training on all media when testing with respect to a particular medium (FNN system).

6.3 Mimicking each Particular Source

In the experiments above, we have been trying to predict whether a sentence is check-worthy in general, i.e., with respect to at least one source; this is how we trained and this is how we evaluated our models. Here, we want to evaluate how well our models perform at finding sentences that contain claims that would be judged as worthy for fact-checking with respect to each of the individual sources. The purpose is to see to what extent we can make our system potentially useful for a particular medium.

Another interesting question is whether we should use our generic system or we should retrain with respect to the target medium. Table 7 shows the results for such a comparison, and it further compares to *CB Platform*. We can see that for all nine media, our model outperforms *CB Platform* in terms of MAP and P@50; this is also true for the other measures in most cases.

Moreover, we can see that training on all data is generally preferable to training on the target medium only, which shows that despite the sizable disagreement between the different media, they do follow some common principles for selecting what is check-worthy; this means that a general system could serve journalists in all these nine, and possibly other, media. One exception is Washington Post, where our system performs better when trained only on the single source, which is an indicator of the difference between Washington Post and the rest sources. Overall, our model works best on PolitiFact, which is a reputable source with fact checking as their primary expertise. We also do well on NPR, NYT, Guardian, and FactCheck; this is quite encouraging.

7 Conclusions and Future Work

We have developed a novel approach for automatically finding worth-checking claims in political debates, which is an understudied problem, despite its importance. Unlike previous work, which has looked primarily at sentences in isolation, here we have focused on the context: relationship between the target statement and the larger context of the debate, interaction between the opponents, and reaction by the moderator and by the public.

Our models have achieved state-of-the-art results, outperforming a strong rivaling system by a margin, while also confirming the importance of the contextual information. We further compiled, and we are making freely available, a new corpus of manually-annotated claims, extracted from the 2016 US presidential and vice-presidential debates, which we gathered from nine reputable sources including FactCheck, PolitiFact, CNN, NYT, WP, and NPR.

In future work, we plan to extend our corpus with additional debates, e.g., from other elections, but also with interviews and general discussions. We would also like to experiment with distant supervision, which would allow us to gather more training data, thus enabling deep learning. We further plan to extend our system with finding claims at the sub-sentence level, as well as with automatic fact-checking of the identified claims.

References

Mouhamadou Lamine Ba, Laure Berti-Equille, Kushal Shah, and Hossam M Hammady. 2016. VERA: A platform for veracity estimation over web data. In *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, WWW '16 Companion, pages 159–162.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on Twitter. In *Proceedings of the 20th international conference on World wide web*. ACM, New York, NY, USA, WWW '11, pages 675–684.

Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M. Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational fact checking from knowledge networks. *PLOS ONE* 10(6):1–13.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2009. Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering* 15(4):i–xvii.

William Davies. 2016. The age of post-truth politics. *New York Times* 24.

Rob Ennals, Dan Byler, John Mark Agosta, and Barbara Rosario. 2010a. What is disputed on the web? In *Proceedings of the 4th workshop on Information credibility*. ACM, New York, NY, USA, WICOW '10, pages 67–74.

Rob Ennals, Beth Trushkowsky, and John Mark Agosta. 2010b. Highlighting disputed claims on the web. In *Proceedings of the 19th international conference on World wide web*. ACM, pages 341–350.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudk, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Fort Lauderdale, FL, USA, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323.

Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2016. In search of credible news. In *Proceedings of the 17th International Conference on Artificial Intelligence: Methodology, Systems, and Applications*. Varna, Bulgaria, AIMSA '16, pages 172–180.

Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. CIKM '15, pages 1835–1838.

Naeemul Hassan, Mark Tremayne, Fatma Arslan, and Chengkai Li. 2016. Comparing automated factual claim detection against judgments of journalism organizations. In *Computation + Journalism Symposium*. Stanford, California, USA.

Joan B. Hooper. 1974. *On Assertive Predicates*. Indiana University Linguistics Club. Indiana University Linguistics Club.

Ken Hyland. 1998. *Hedging in scientific research articles*, volume 54. John Benjamins Publishing.

Shafiq Joty, Giuseppe Carenini, and Raymond T. Ng. 2015. CODRA: A novel discriminative framework for rhetorical analysis. *Comput. Linguist.* 41(3):385–435.

Georgi Karadzhov, Pepa Gencheva, Preslav Nakov, and Ivan Koychev. 2017a. We built a fake news & click-bait filter: What happened next will blow your mind! In *Proceedings of the 2017 International Conference on Recent Advances in Natural Language Processing*. Varna, Bulgaria, RANLP '17.

Georgi Karadzhov, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cede no, and Ivan Koychev. 2017b. Fully automated fact checking using external sources. In *Proceedings of the 2017 International Conference on Recent Advances in Natural Language Processing*. Varna, Bulgaria, RANLP '17.

Lauri Karttunen. 1971. Implicative verbs. *Language* pages 340–358.

Dieu-Thu Le, Ngoc Thang Vu, and Andre Blessing. 2016. Towards a text analysis system for political debates. *LaTeCH 2016* page 134.

Yann LeCun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*. Anchorage, Alaska, USA, volume 86 of *1998 IEEE*, pages 2278–2324.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*. New York, NY, USA, WWW '05, pages 342–351.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*. Philadelphia, Pennsylvania, ETMTNLP '02, pages 63–70.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. New York, New York, USA, IJCAI'16, pages 3818–3824.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR* abs/1309.4168.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon 29(3):436–465.

Preslav Nakov, Tsvetomila Mihaylova, Lluís Màrquez, Yashkumar Shiroya, and Ivan Koychev. 2017. Do not trust the trolls: Predicting credibility in community question answering forums. In *Proceedings of the 2017 International Conference on Recent Advances in Natural Language Processing*. Varna, Bulgaria, RANLP '17.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. Sofia, Bulgaria, volume 1 of *ACL '13*, pages 1650–1659.

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