To Check or not to Check: Syntax, Semantics, and Context in the Language of Check-worthy Claims

(Best of the Labs Track at CLEF-2019)

Chaoyuan Zuo[1], Ayla Ida Karakas[2], and Ritwik Banerjee[1]

[1] Department of Computer Science
{chzuo,rbanerjee}@cs.stonybrook.edu
[2] Department of Linguistics
ayla.karakas@stonybrook.edu
Stony Brook University, Stony Brook New York 11794, USA

Abstract. As the spread of information has received a compelling boost due to pervasive use of social media, so has the spread of misinformation. The sheer volume of data has rendered the traditional methods of expert-driven manual fact-checking largely infeasible. As a result, computational linguistics and data-driven algorithms have been explored in recent years. Despite this progress, identifying and prioritizing *what* needs to be checked has received little attention. Given that expert-driven manual intervention is likely to remain an important component of fact-checking, especially in specific domains (e.g., politics, environmental science), this identification and prioritization is critical. A successful algorithmic ranking of "check-worthy" claims can help an expert-in-the-loop fact-checking system, thereby reducing the expert's workload while still tackling the most salient bits of misinformation. In this work, we explore how linguistic syntax, semantics, and the contextual meaning of words play a role in determining the check-worthiness of claims. Our preliminary experiments used explicit stylometric features and simple word embeddings on the English language dataset in the Check-worthiness task of the CLEF-2018 Fact-Checking Lab, where our primary solution outperformed the other systems in terms of the mean average precision, R-precision, reciprocal rank, and precision at k for multiple values k. Here, we present an extension of this approach with more sophisticated word embeddings and report further improvements in this task.

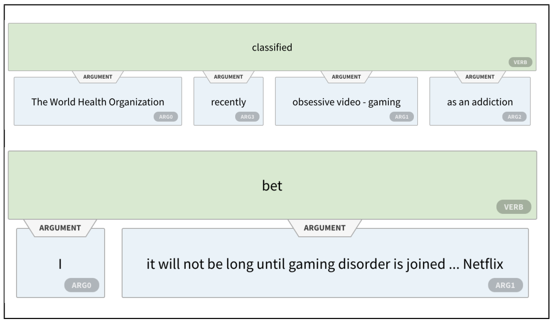
Keywords: Check-worthiness · Multi-layer Perceptron · SVM · Word Embedding · Context · Syntax · Semantics

1 Introduction

We live in an age where a significant part of our lives may be infused with the information we see on the web and social media platforms, and most Internet users readily acquire the power of civilian commentary [2,13]. Thus, any information

Fig. 1. A two-sentence excerpt from an article about Netflix addiction and the main pieces of information extracted by Open IE [34]. A pipeline of such tools is ill-suited for fact-checking since there is no distinction between check-worthy and other events. Here, only the first sentence is worth verifying, and there is no need to fact-check the author's bet. Source: https://www.theguardian.com/commentisfree/2018/jun/20/netflix-addiction-is-real-we-are-entertaining-ourselves-to-death (accessed May 27, 2019).

The World Health Organization recently classified obsessive video-gaming as an addiction. I bet it will not be long until "gaming disorder" is joined in the WHO's International Classification of Diseases by another modern, screen-based malady: "Netflix disorder".



available on the web has the potential to be disseminated with unprecedented speed and scope. As a result, the ordinary web or social media user is often subjected to information overload [33]. Given how onerous the task of fact-checking can become under vast amounts of information, users often resort to confirmation bias when it comes to selection and retention of information [22]. Such a confluence of information overload and bias can create a society where unverified claims can easily masquerade as facts. We may not be able to overcome individual biases purely with technology. We can, however, mitigate the ill-effects of misinformation by identifying and prioritizing *what* needs to be verified. For small snippets, this is obvious to the human reader but beyond the scope of existing tools, as illustrated by the excerpt in Fig. 1. Prior work on automatic fact-checking cannot, however, make this distinction, and need to extract statements to be fact-checked before the actual verification process can begin [4,5,17,36,38].

An evident benefit of prioritizing check-worthy statements is that the ranked list can then be provided to journalists and users to help them focus on verifying the most important statements without information overload. Moreover, accurate identification of check-worthy statements will clearly benefit any subsequent fact-checking processes. With this as our motivation, we focus on the task of identifying and ranking statements based on their check-worthiness, as defined by the CLEF 2018 Fact Checking Lab [1]. Here, we present our work using the data and evaluation framework provided by the CLEF 2018 Lab on Automatic Identification and Verification of Claims in Political Debates [27], and present a solution that significantly outperforms other systems in terms of several performance measures.

The remainder of this paper is organized into a discussion of related work in Section 2, a description of our methodology in Section 3, followed by the experiments and their evaluation in Section 4. Then in Section 5 we provide a detailed analysis of our results before finally concluding with some possible directions for future work in Section. 6.

2 Related Work

The importance of accurate information has been widely appreciated in many distinct but increasingly interconnected fields, from journalism to social psychology to computational linguistics. Computational approaches to fact-checking are intended to overcome the hurdle of verifying large amounts of information, and have become a fundamental component of computational journalism due to its critical role in upholding the accuracy and verifiability of information [5,7,10].

Very recently, some work has also been done on the extraction of numerical and statistical claims to identify whether or not a statement is worth verifying [3]. This, however, is orthogonal to our exploration of check-worthiness based on linguistic constructs, where early research focused on general misinformation such as rumors and hoaxes [23,31], and soon, several automatic fact-checking systems were developed for the verification of political news [17,36]. The preliminary step of identifying *what* is worth checking, however, received little attention until 2015 (ClaimBuster [16]), and it was not until its later incarnation that the identification of check-worthiness was incorporated into a fact-checking system [17].

Initially, ClaimBuster modeled the identification of check-worthy statements as a three-class supervised classification task where each sentence had one of three possible labels: check-worthy factual, unimportant factual, and non-factual. They used a dataset of political debates by U.S. presidential candidates, and crowdsourced the sentence labels. A small set of lexical (words and sentence length), shallow syntactic (part-of-speech tags), and semantic (sentiment and entity types) features were extracted and filtered through a random forest classifier for selection. Later, each sentence was assigned a score based on a classification and scoring model, with similar features as before [17]. Another notable work on check-worthiness treated it as a ranking task [11] using a richer set of features. This work also released the CW-USPD-2016 dataset [11], which comprises four political debates. Soon, a larger dataset was built with fifteen additional political debates [29], and a multi-classifier named TATHYA was designed to identify the check-worthy statements in this dataset. This generated a topic probability distribution together with the use of lexical and shallow syntactic features, and provided a comparison against ClaimBuster on the test set. Both systems, however, reported relatively low F1-scores of 0.179 and 0.214, respectively.

Evidently, there is scope for improvement in identifying and ranking check-worthy statements. With the above-mentioned body of work as the foundation, and based on observations regarding the extent of overlap between lexical and shallow syntactic features [11,24], we use a significantly richer set of features including clause and phrase structures and word embeddings to account for syntax, semantics, and linguistic context.

3 Methodology

In this section we begin with a description of the data and then proceed to describe the feature selection and the use of algorithmic heuristics. At this point, we would like to underscore that our long-term goal is to aid fact-checking in a domain-independent manner. Thus, we have chosen to work purely with linguistic constructs, without explicit external domain knowledge.

3.1 Data

We use the English language political debate dataset provided as part of the CLEF 2018 Check-Worthiness task [1]. The training data consists of three political debates, where each sentence is associated with its speaker and expert-annotated as check-worthy or not. It is a highly imbalanced dataset where out of the total 3,989 sentences, only 94 (2.36%) are labeled check-worthy.

The test data consists of two debates and five speeches with 2,815 and 2,064 sentences, respectively[1]. In this dataset, a speaker is associated with every sentence, but they are not normalized. For example, it includes "Hillary Clinton (D-NY)", "Former Secretary of State, Presidential Candidate", and simply "Clinton". Since these are all referring to the same speaker, we map these variations to a single entity. The training set comprises only debates where multiple entities frequently engage in a conversation. The test set, however, contains speeches by each presidential candidate. The rhetorical structure of a debate, of course, is very different from a speech. We thus extract all the sentences attributed to a particular entity and create entity-specific sub-datasets, which serve to train models to identify check-worthy sentences from speeches. For debates, on the other hand, we use the original training data to train the models.

3.2 Feature Design & Selection

For lexical features, we remove function words and stem the content words using the Snowball stemmer [30]. The remaining set of features we explore can be broadly grouped into (a) syntactic or stylometric, (b) semantic, and (c) contextual.

Syntactic Features. This category may further be divided into shallow and deep syntactic features. Following Gencheva et al. [11], we extract the following shallow syntactic features from a sentence: part-of-speech tags, total number of tokens in a sentence, number of negations, and number of tokens in the past, present, and future tenses. The more complex patterns of language and how they may serve as cues to deceptive statements cannot, however, be understood based on these features [8]. To this end, we generate the constituency parse tree for every sentence, and extract the clause-level and phrase-level tags.

Semantic Features. We use the Stanford named entity recognition system [9] to extract the number of named entities in each sentence. We also distinguish between mentions of people and other entity types by appending an extra feature for entities recognized as the type person. To capture even more meaning from

[1] The dataset does not provide this categorization, but we treat them differently since a *debate*, unlike a *speech*, has interactive discourse between multiple speakers.

each sentence, we also extract “affect” (direct sentiment as well as connotation). For sentiment scoring, we use the TextBlob [25] library to train a naïve Bayes classifier on the well known movie review corpus [28]. To extract the use of more subtle semantics, we use markers for connotation, subjectivity, bias, and opinion. While the last three have been used in earlier work [11], our approach is the first to incorporate connotation features for check-worthiness. We use Connotation WordNet [20] to obtain a score for each word, and for a sentence, we compute the mean of the connotation scores of its words. For subjectivity and bias, we use lexicons to obtain information about the extent to which each word is used to (a) voice a subjective notion [37], (b) make biased statements [32], and (c) share positive or negative opinions [19]. For each sentence, we thus have four new features beyond overt sentiment – connotation, subjectivity, bias, and opinion – with each feature's value simply being the total score in the corresponding lexicon of all the words in that sentence.

Contextual Features. When incorporating context, we resort to explicit feature extraction as well as word embeddings.

The explicit features comprise metadata and discourse. We use three binary metadata features, indicating whether or not (i) the speaker is the anchor/host, (ii) the speaker's opponent is mentioned in the sentence, and (iii) the sentence is immediately followed by an intense reaction from the audience, which is encoded in the training data as a ‘system’ reaction. All the features described so far make no distinction between speeches and debates. However, since debates have an interactive discourse structure, we identify every “segment” in the data, defined to be a maximal set of consecutive sentences by the same speaker [11]. As discourse features, we use the relative position of a sentence within its segment, and also the number of sentences in the previous, current, and the following segments.

Beyond explicit features, we use word and sentence embeddings to model context. Traditional vector space modeling associates each dimension with a word, which does not account for the distributional hypothesis in linguistics: words occurring in similar contexts tend to have similar meanings [15]. In contrast, recent work has shown that using the embedding of a word in a latent semantic space often leads to better results in various tasks [6,26] since such embeddings try to account for the word's context in various ways. Inspired by this approach to capturing context, we use word embeddings such that a whole sentence is represented by a vector in a low-dimensional space where similar meanings are closer together. Since this is a topic of extensive research in its own right, we make use of readily available embeddings.

The first approach is to use the 300-dimensional pre-trained Google News word embeddings developed by Mikolov et al. [26], and take the arithmetic mean of all the word vectors to obtain a representation for the entire sentence. The second is to use the state-of-the-art BERT [6] embeddings to obtain sentence-level encoding through their service API [39].

Feature Selection. All the above features result in a very high-dimensional feature space, which is known to decrease the predictive power of models [35]. Especially in this work, the extreme class imbalance (see Section. 3.1) could mean that features indicative of the minority label get ignored by the classifier. Therefore, we apply a two-stage dimensionality reduction process for the combination of all the features described earlier, except for BERT embeddings. This two-stage feature selection together with the sampling process is novel for this task.

First, we perform univariate feature selection using the χ^2-test and retain the 2,000 best features. Second, we exploit the property that linear predictive models with L1 loss lead to sparse solutions and encourage the vanishing coefficients for weakly correlated features. We do this first for the entire training data, and then with repeated undersampling for the majority class. This way, after each iteration we obtain a small but balanced training set. For every sample generated in this manner, we train a L1-regularized support vector machine (SVM) classifier and discard the features with vanishing coefficients. At the end of this, we are left with a space of 2,655 and 2,404 features for debates and speeches, respectively.

For BERT sentence embeddings, which are provided as pre-trained continuous space language models for text representation, we avoid the χ^2-test because BERT was designed to capture contextual linguistic information and the individual dimensions do not represent separate variables in the traditional sense. Its training process captures deep bidirectional representations from enormous amounts of text in an unsupervised fashion. allowing the embedding to implicitly capture both the left and right context of words in a sentence. We do however, use SVM with L1 regularization as above, resulting in 306 and 260 features for debates and speeches, respectively.

3.3 Classification Models

We use two supervised learning algorithms, SVM and multilayer perceptrons (MLP). Here, we describe these models along with their training processes.

As discussed in Section 3.2, we use SVM with L1 regularization for feature selection. However, since the L1 loss function suffers from a tendency to miss optimal solutions, we use L2 loss to build the final model after feature selection. Our MLP uses two hidden layers with 100 units and 8 units, respectively. We use the hyperbolic tangent (tanh) as the activation function, as it achieved better results when compared to rectified linear units (ReLU). Stochastic optimization is done with Adam [21]. Finally, to avoid overfitting, we use L2-regularization in both SVM and MLP. The regularization parameter of SVM is set to 0.02 for debates and 0.01 for speeches. For both learners, the extreme class imbalance was an obstacle, and we use the adaptive synthetic sampling algorithm ADASYN [18] to overcome it. Additionally, we also build an ensemble model combining the SVM and MLP with equal weights. In this model, we provide the final score for each sentence as a normalization by standard deviation of the results of SVM and MLP followed by computing the average.

We use 3-fold cross-validation to select the final model for debates, using two files for training and the remaining one for testing, performance evaluation, and parameter tuning. For speeches, we split the training samples into two halves (one file in each) for 2-fold cross-validation.

Algorithm 1 Heuristics for assigning the check-worthiness score $w(\cdot)$ to sentences. The minimum token count constants were experimentally chosen by measuring the cross-validation performance on the training data. The default minimum score of 10^{-8} was used simply to provide a very low non-zero score for sentences that are highly unlikely to be check-worthy.

```
Require: category $\in \{\text{SPEECH}, \text{DEBATE}\}$,
  strict_mode $\in$ {true, false}, sentence $S$.

  MIN_TOKEN_COUNT $\leftarrow 0$
  if category is SPEECH then
    if strict_mode then
      MIN_TOKEN_COUNT $\leftarrow 10$
    else
      MIN_TOKEN_COUNT $\leftarrow 8$
    end if
  else
    if strict_mode then
      MIN_TOKEN_COUNT $\leftarrow 7$
    else
      MIN_TOKEN_COUNT $\leftarrow 5$
    end if
  end if
  if $S_{\text{SPEAKER}}$ is SYSTEM then
    $w(S) \leftarrow 10^{-8}$
  end if
  if $S_{\text{NUMBER OF TOKENS}} <$ MIN_TOKEN_COUNT
  then
    $w(S) \leftarrow 10^{-8}$
  end if
  if $S$ contains "thank you" then
    $w(S) \leftarrow 10^{-8}$
  end if
  if $S_{\text{NUMBER OF SUBJECTS}} < 1$ then
    if category is SPEECH then
      $w(S) \leftarrow 10^{-8}$
    else if $S$ contains "?" then
      $w(S) \leftarrow 10^{-8}$
    end if
  end if
```

3.4 Heuristics

As a final step of our approach, we introduce a few simple heuristics to override the scores assigned by the MLP models. These rules differ slightly based on (i) the category, i.e., speech or debate, and (ii) whether or not 'strict' (a flag to control the threshold sentence size) heuristics are being used – when active, it tends to discard more sentences. The complete set of heuristics are specified in Algorithm 1. One particular rule requires the identification of nominal subjects in a sentence. To extract this, we generate dependency parse trees of the sentence and count the number of times any of the following dependency labels appear: `nsubj`, `csubj`, `nsubjpass`, `csubjpass`, or `xsubj`. The first two indicate nominal and clausal subjects, respectively. The next two indicate nominal and clausal subjects in a passive clause, and the last label denotes a controlling subject, which relates an open clausal complement to its external clause.

4 Experiments and Results

The primary evaluation measure in the CLEF 2018 task was the mean of average precisions over all queries, or mean average precision (MAP), where average precision is defined in terms of the number of check-worthy sentences n_{chk}, the total number of sentences n, the precision $\text{Prec}(k)$ at cut-off k in the list of

Table 1. Primary submission results of the top three teams, based on mean average precision (MAP). The mean reciprocal rank (MRR) and mean precision at rank k (MP@k) evaluation measures are also shown.

TEAM	**MAP**	**MRR**	**MRP**	**MP@1**	**MP@3**	**MP@5**	**MP@10**	**MP@20**	**MP@50**
Zuo *et al.* [40]*	**0.1332**	**0.4965**	**0.1352**	**0.4286**	**0.2857**	0.2000	0.1429	**0.1571**	0.1200
Hansen *et al.* [14]	0.1152	0.3159	0.1100	0.1429	0.1429	0.1143	0.1286	0.1286	**0.1257**
Ghanem *et al.* [12]	0.1130	0.4615	0.1315	0.2857	0.2381	**0.3143**	**0.2286**	0.1214	0.0866

sentences ranked by check-worthiness, and the indicator function $\delta(k)$ equaling 1 if the sentence at rank k is check-worthy, and 0 otherwise:

$$AP = \frac{1}{n_{\text{chk}}} \sum_{k=1}^{n} \text{Prec}(k) \cdot \delta(k).$$

As part of the CLEF 2018 Check-Worthiness task, we submitted the MLP model without the strict heuristics as our primary run. The results of the top three teams and their primary runs are shown in Table 1, where our model [40] achieved the best performance with respect to multiple evaluation criteria. In particular, our approach performed the best in terms of the primary evaluation measure MAP, with a score of 0.1332. Our primary submission also had the best performance with regard to placing the most check-worthy statements at the very top of the ranked list, as shown by the mean precision at lower ranks, MP@1 and MP@3. We submitted two contrastive runs as well, one with the strict heuristics and another with the ensemble model where the MLP component was without the strict heuristics. Additionally, we also tested the MLP model with no heuristics at all. For all these models, we used the hand-crafted linguistic features described in Section 3.2 and the Google News word embeddings. Our experiments with BERT embeddings were driven to test whether or not they already capture contextual information better than the hand-crafted features. To explore this, we used both MLP and SVM with and without (a) the handcrafted linguistic features, and (b) the feature selection steps.

The performance of these models on the test set are shown in Table 2. For the sake of brevity, we have omitted a few other combinations of feature selection and heuristics. One of our constractive runs, MLP$_{\text{str}}$, performed even better on the test set than the primary submission. The overall best performance is achieved by SVM with BERT embeddings where feature selection (from the sentence embedding vectors) is done using L1 regularization, but no hand-crafted features are used. For this model, we also report our results on the speeches and debates separately in Table 3.

5 Error Analysis and Conjectures

Identifying and prioritizing check-worthy sentences is clearly a difficult task, with even the best model suffering from a rather large number of incorrect

Table 2. Model performance on the test set. *With Google News word embeddings*: MLP*, MLP$_{\text{str}}$, and MLP$_{\text{none}}$ (strict, non-strict, and no heuristics); ENS (ensemble model). MLP* was our primary submission, and MLP$_{\text{str}}$ and ENS were the two contrastive submissions in the CLEF 2018 Lab task. *With BERT embeddings*: the subscript $\pm\mathcal{L}$ denotes whether or not the handcrafted linguistic features are used, and the superscript (†) denotes that feature selection was not performed.

	MAP	**MRR**	**MRP**	**MP@1**	**MP@3**	**MP@5**	**MP@10**	**MP@20**	**MP@50**
With Google News embeddings:									
MLP*	0.1332	0.4965	0.1352	0.4286	0.2857	0.2000	0.1429	0.1571	0.1200
MLP$_{\text{str}}$	0.1366	0.5246	0.1475	0.4286	0.2857	0.2286	0.1571	0.1714	0.1229
MLP$_{\text{none}}$	0.1086	0.4767	0.1037	0.2857	0.2857	0.2000	0.1286	0.1071	0.1000
ENS	0.1317	0.4139	0.1523	0.2857	0.1905	0.1714	0.1571	0.1571	0.1429
With BERT embeddings:									
MLP$_{-\mathcal{L}}$	0.1499	0.4931	0.1601	0.2857	0.3333	0.2571	0.1571	0.1857	0.1314
SVM$^{(\dagger)}_{+\mathcal{L}}$	0.1648	0.3967	0.2170	0.1429	0.2381	0.2286	0.2714	0.20171	0.1600
SVM$^{(\dagger)}_{-\mathcal{L}}$	0.1770	0.5276	0.2002	0.2857	0.2381	0.2286	0.2286	0.2643	0.1486
SVM$_{+\mathcal{L}}$	0.1760	0.4396	0.2157	0.1429	0.2857	0.2571	0.2429	2071	0.1742
SVM$_{-\mathcal{L}}$	**0.1974**	**0.7508**	**0.2201**	**0.7143**	**0.3333**	**0.2857**	**0.2286**	**0.2571**	**0.1657**

classifications. In this section, we analyze the models explored, and present our conjectures regarding why they may be failing to handle many sentences.

First, we observe that tense plays a logically consistent role in check-worthiness – future actions cannot be verified, of course. POS tagging, however, often confuses the future tense with political promises made using the present continuous (e.g., "We're cutting taxes."). We also observe that anecdotes are often wrongly prioritized as check-worthy. These sentences are usually complex, with a lot of content, which makes it easy for the model to conflate them with other complex sentences about check-worthy real events. The dataset contains a few duplicate sentences as well as very similar sentences with different labels, which could be amplifying errors or discarding potentially useful features.

Rhetorics, too, plays a critical role. They often break the structures associated with standard sentence formation. Several incorrect predictions were due to constructs such as *scesis onomaton*, where words or phrases with nearly equivalent meaning are repeated. This could make the model falsely believe that there is more informational content in the sentence. This is even harder to handle with multiple speakers in debates. The conversational aspect also causes another problem: quite a few sentences are short, and in isolation, would not be check-worthy. However, as a response to things mentioned earlier in the debate, they are. A related issue is the use of sentence fragments. This is sparingly used in the literature to intentionally create dramatic effect, but was seen frequently in the debates due to the prevalence of ill-formed or partly-formed sentences stopping and giving way to another sentence. In some cases, the fragments verbal repetitions, *e.g.*, "Ambassador Stevens – Ambassador Stevens sent 600 requests for help." In light of this analysis, we believe that our features are better suited for formal written language than speech or debate transcripts.

Table 3. Performance of the best performing model $\text{SVM}_{-\mathcal{L}}$ (SVM with BERT embeddings and no hand-crafted features) on debates and speeches from the test set.

	MAP	**MRR**	**MRP**	**MP@1**	**MP@3**	**MP@5**	**MP@10**	**MP@20**	**MP@50**
Debates	0.1185	0.5278	0.1234	0.5000	0.3333	0.**3000**	**0.3000**	0.**2750**	**0.1500**
Speeches	**0.2290**	**0.8400**	**0.2588**	**0.8000**	**0.3333**	0.2800	0.1400	0.1600	0.1040

Overall, the SVM model using only BERT embeddings achieved the best performance. This may seem somewhat surprising, but given that the deep bidirectional network implicitly captures a lot of contextual information (unlike older methods like the Google News embeddings). It is worth noting, however, that feature selection played an important role across all models. With BERT embeddings, we observe that while it performs significantly better on many sentences, if often continues to fail in the presence of complex syntactic structures and frequent use of named entities. This is quite possibly because the context captured by these embeddings regard a sentence as a flat structure, which works reasonably well until the parse trees become deeper. Another potential reason for failure could be that the pre-trained embeddings are derived from vast amounts of data where the named entities (*e.g.*, Iraq, Donald, ISIS) are, indeed, correlated with check-worthy statements whereas in the data used in this work, these entities also frequently appear in sentences that are not worth checking.

Just like the hand-crafted linguistics features and Google News word embeddings, the BERT embeddings perform poorly on debate transcripts (shown explicitly in Table 3 for the best model).

6 Conclusion and Future Work

We described several models to detect check-worthy sentences in political debates and speeches. Some models combine a few rules with supervised learning using linguistic features, whereas others use contextual embeddings. This work opens up several possibilities for future research in the field of fact-checking. First, we would like to perform ablation tests to confirm some of the anecdotal observation and conjectures made in the previous section. Next, we would also like to study in greater detail the linguistic forms of *informational* content. This has been done qualitatively in sociolinguistics, and some work has even looked into deep syntactic features, but it has not yet been applied to identifying check-worthy sentences. Furthermore, the use of contextual embeddings show that even with deep learning, there is significant scope for improvement in terms of modeling context in word and sentence representations. Along this line, we plan to investigate methods where context is explored in conjunction with deep syntactic structures to overcome the current deficiencies in identifying check-worthy sentences. In order for this line of research to impact journalists and citizens, there is also a need for complementary thrusts in related areas such as social network analysis, information source identification, and trustworthy crowdsourcing.

References

1. Atanasova, P., Màrquez, L., Barrón-Cedeño, A., Elsayed, T., Suwaileh, R., Zaghouani, W., Kyuchukov, S., Da San Martino, G., Nakov, P.: Overview of the CLEF-2018 CheckThat! Lab on Automatic Identification and Verification of Political Claims, Task 1: Check-Worthiness. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) CLEF 2018 Working Notes (2018)
2. Bruns, A., Highfield, T.: Blogs, Twitter, and breaking news: the produsage of citizen journalism. In: Produsing Theory in a Digital World: The Intersection of Audiences and Production in Contemporary Theory, vol. 80, pp. 15–32. Peter Lang (2012)
3. Cao, T.D., Manolescu, I., Tannier, X.: Extracting statistical mentions from textual claims to provide trusted content. In: NLDB (2019)
4. Cazalens, S., Lamarre, P., Leblay, J., Manolescu, I., Tannier, X.: A content management perspective on fact-checking. In: " Journalism, Misinformation and Fact Checking" alternate paper track of" The Web Conference" (2018)
5. Cohen, S., Li, C., Yang, J., Yu, C.: Computational Journalism: A Call to Arms to Database Researchers. In: Conference on Innovative Data Systems Research. CIDR '11, ACM, Asilomar, California, USA (2011)
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018)
7. Diakopoulos, N.: A Functional Roadmap for Innovation in Computational Journalism. Tech. rep., Rutgers University (2011)
8. Feng, S., Banerjee, R., Choi, Y.: Syntactic Stylometry for Deception Detection. In: ACL (2). pp. 171–175 (2012)
9. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: ACL. pp. 363–370 (2005)
10. Flew, T., Spurgeon, C., Daniel, A., Swift, A.: The promise of computational journalism. Journalism Practice **6**(2), 157–171 (2012)
11. Gencheva, P., Nakov, P., Màrquez, L., Barrón-Cedeño, A., Koychev, I.: A context-aware approach for detecting worth-checking claims in political debates. In: RANLP 2017. pp. 267–276 (2017)
12. Ghanem, B., Montes-y Gómez, M., Rangel, F., Rosso, P.: UPV-INAOE-Autoritas-Check That: Preliminary Approach for Checking Worthiness of Claims. In: CLEF Working Notes (2018)
13. Goode, L.: Social news, citizen journalism and democracy. New media & society **11**(8), 1287–1305 (2009)
14. Hansen, C., Hansen, C., Simonsen, J.G., Lioma, C.: The Copenhagen Team Participation in the Check-Worthiness Task of the Competition of Automatic Identification and Verification of Claims in Political Debates of the CLEF-2018 CheckThat! Lab. In: CLEF Working Notes (2018)
15. Harris, Z.S.: Distributional Structure. Word **10**(2-3), 146–162 (1954)
16. Hassan, N., Li, C., Tremayne, M.: Detecting check-worthy factual claims in presidential debates. In: CIKM. pp. 1835–1838. CIKM (2015)
17. Hassan, N., Zhang, G., Arslan, F., Caraballo, J., Jimenez, D., Gawsane, S., Hasan, S., Joseph, M., Kulkarni, A., Nayak, A.K., et al.: ClaimBuster: The First-ever End-to-end Fact-checking System. Proceedings of the VLDB Endowment **10**(12), 1945–1948 (2017)
18. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In: Proceedings of the IEEE Joint Conference on Neural Networks (IJCNN), 2008. pp. 1322–1328. IEEE (2008)

19. Hu, M., Liu, B.: Mining and Summarizing Customer Reviews. In: ACM SIGKDD. pp. 168–177. ACM (2004)
20. Kang, J.S., Feng, S., Akoglu, L., Choi, Y.: ConnotationWordNet: Learning Connotation over the Word+Sense Network. In: ACL. pp. 1544–1554. Association for Computational Linguistics (June 2014)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
22. Klayman, J.: Varieties of Confirmation Bias. In: Psychology of learning and motivation, vol. 32, pp. 385–418. Elsevier (1995)
23. Kumar, S., West, R., Leskovec, J.: Disinformation on the Web: Impact, Characteristics, and Detection of Wikipedia Hoaxes. In: Proc. 25th Int. Conf. World Wide Web. pp. 591–602. Int. WWWW Conference Committee (IW3C2) (2016)
24. Le, D.T., Vu, N.T., Blessing, A.: Towards a text analysis system for political debates. In: Proceedings of the 10^{th} SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities. pp. 134–139 (2016)
25. Loria, S.: TextBlob: Simplified Text Processing. http://textblob.readthedocs.org/en/dev/ (2014)
26. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781 (2013)
27. Nakov, P., Barrón-Cedeño, A., Elsayed, T., Suwaileh, R., Màrquez, L., Zaghouani, W., Gencheva, P., Kyuchukov, S., Da San Martino, G.: Overview of the CLEF-2018 Lab on Automatic Identification and Verification of Claims in Political Debates. In: Working Notes of CLEF 2018 – Conference and Labs of the Evaluation Forum. CLEF '18, Avignon, France (September 2018)
28. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: EMNLP. pp. 79–86 (2002)
29. Patwari, A., Goldwasser, D., Bagchi, S.: TATHYA: A Multi-Classifier System for Detecting Check-Worthy Statements in Political Debates. In: CIKM. pp. 1–4 (2017)
30. Porter, M.F.: Snowball: A Language for Stemming Algorithms. http://snowball.tartarus.org/texts/introduction.html (2001)
31. Qazvinian, V., Rosengren, E., Radev, D., Mei, Q.: Rumor has it: Identifying Misinformation in Microblogs. In: EMNLP. pp. 1589–1599. ACL (2011)
32. Recasens, M., Danescu-Niculescu-Mizil, C., Jurafsky, D.: Linguistic Models for Analyzing and Detecting Biased Language. In: ACL. vol. 1, pp. 1650–1659 (2013)
33. Rodriguez, M.G., Gummadi, K., Schoelkopf, B.: Quantifying Information Overload in Social Media and Its Impact on Social Contagions. In: ICWSM (2014)
34. Stanovsky, G., Michael, J., Zettlemoyer, L., Dagan, I.: Supervised Open Information Extraction. In: NAACL-HLT, Vol. 1 (Long Papers). pp. 885–895 (2018)
35. Trunk, G.V.: A Problem of Dimensionality: A Simple Example. IEEE Transactions on Pattern Analysis and Machine Intelligence **1**(3), 306–307 (1979)
36. Vlachos, A., Riedel, S.: Fact Checking: Task definition and dataset construction. In: Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science. pp. 18–22 (2014)
37. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In: EMNLP. pp. 347–354 (2005)
38. Wu, Y., Agarwal, P.K., Li, C., Yang, J., Yu, C.: Toward computational fact-checking. Proceedings of the VLDB Endowment **7**(7), 589–600 (2014)
39. Xiao, H.: bert-as-service. https://github.com/hanxiao/bert-as-service (2018)
40. Zuo, C., Karakas, A., Banerjee, R.: A Hybrid Recognition System for Check-worthy Claims Using Heuristics and Supervised Learning. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) CLEF 2018 Working Notes (2018)