CEUR-WS.org/Vol-2125/paper_143.pdf

A Hybrid Recognition System for Check-worthy Claims Using Heuristics and Supervised Learning

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. In recent years, the speed at which information disseminates has received an alarming boost from the pervasive usage of social media. To the detriment of political and social stability, this has also made it easier to quickly spread false claims. Due to the sheer volume of information, manual fact-checking seems infeasible, and as a result, computational approaches have been recently explored for automated fact-checking. In spite of the recent advancements in this direction, the critical step of recognizing and prioritizing statements worth fact-checking has received little attention. In this paper, we propose a hybrid approach that combines simple heuristics with supervised machine learning to identify claims made in political debates and speeches, and provide a mechanism to rank them in terms of their "check-worthiness". The viability of our method is demonstrated by evaluations on the English language dataset as part of the Check-worthiness task of the CLEF-2018 Fact Checking Lab.

Keywords: Check-worthiness · Multi-layer Perceptron · Heuristics · Feature Selection · Stylometry

1 Introduction

It is no secret that we live in an age of ubiquitous web and social media. For the most part, any Internet user readily acquires the latent power of civilian commentary and journalism [3,10]. Consequently, information available on the web now carries the potential to propagate amid the public domain with unprecedented speed and reach. The ordinary Internet user, however, contends with an overwhelming amount of information, which makes the task of determining the accuracy and integrity of the claims all the more onerous. Additionally, users usually want their beliefs to be confirmed by information [18,34]. The confluence of vast amounts of information and such confirmation bias, thus, can create a society where unverified information runs amok masquerading as facts. While correcting confirmation biases at a social scale may be extremely challenging and even controversial, the spread of misinformation can be mitigated by focusing only on curating the claims.

Comprehensive manual fact-checking is highly tedious and, in light of the sheer volume of information, infeasible. To overcome this hurdle, several approaches to automated fact-checking have been proposed in the nascent field of computational journalism [5,8]. Some prior work took to computing the semantic similarity between claims [4,13], while others proposed fact-checking as a question-answering task [5,33,36]. Both approaches need to extract statements to be fact-checked before the actual verification process can begin. ClaimBuster [12] was the first fact-checking system that assigned to each sentence a check-worthiness score between 0 and 1. Subsequently, a multi-class classification approach with fewer features was explored to specifically identify check-worthy claims, but it suffered from comparatively lower precision [28]. Outside of this small body of work, the preliminary step of identifying check-worthy claims has received little attention. Gencheva et al. [9] were the first to develop a publicly available dataset for this task. Their annotations were obtained from nine fact-checking websites. They also used a significantly richer feature set. Keeping in line with the observations made by prior work regarding the extent of overlap in lexical and shallow syntactic features [9,20], we use a significantly richer set of features derived from word embeddings and deep syntactic structures.

In this work, our focus is on recognizing “check-worthy” statements. Accurate identification of such statements will benefit the fact-checking and verification processes that follow, independent of the specific techniques used therein. We use the task formulation, data, and evaluation framework provided by the CLEF-2018 Lab on Automatic Identification and Verification of Claims in Political Debates [24] as part of their first task – Check-Worthiness [1].

2 Task, Data, and Evaluation Framework

The CLEF 2018 Fact Checking Lab designed two tasks that, when put together, form the complete fact-checking pipeline. In this work, however, we focus exclusively on the first.

2.1 The Task: Check-Worthiness

The first task – check-worthiness – was defined by the CLEF 2018 Fact Checking Lab as follows:

> Predict which claim in a political debate should be prioritized for fact-checking. In particular, given a debate, the goal is to produce a ranked list of its sentences based on their worthiness for fact checking [9].

The goal of this task is to automatically recognize claims worth checking, and present them in order of priority (i.e., as a ranked list of claims) to journalists or even ordinary Internet and social media users. The ranking is attained in terms of a check-worthiness score. This approach helps the recipient tackle the

Table 1. Labeled sentence examples from political debates provided as training data. Check-worthy sentences are labeled 1, and others are labeled 0. Audience reaction and other background noise is encoded as "SYSTEM"-generated.

Speaker	**Sentence**	**Label**
HOLT	I'm Lester Holt, anchor of "NBC Nightly News."	0
HOLT	I want to welcome you to the first presidential debate.	0
TRUMP	Our jobs are fleeing the country.	0
TRUMP	Thousands of jobs leaving Michigan, leaving Ohio.	1
CLINTON	Donald thinks that climate change is a hoax perpetrated by the Chinese.	1
SYSTEM	(*applause*)	0

problem of information overload and instead, directly focus on the most important statements. The output, therefore, can be fed to an automated fact-checker or be used in a manual pursuit of verification. Either way, it can raise awareness of individual users and stymie the dissemination of false claims in social media.

2.2 Data

Given the alleged impact of disinformation and 'fake news' on the 2016 US presidential election, and the controversy surrounding it, any data pertaining to this election cycle is extremely relevant in terms of fact-checking endeavors having a positive social and political impact in the future. As such, a political debate dataset was provided in English and Arabic. Since our methodology involves heuristics that rely on linguistic insight, we used the English language dataset.

The training data comprised three political debates. Each debate was split into sentences, and each sentence was associated with its speaker and annotated by experts as check-worthy or not (labeled 1 and 0, respectively). This data contained a total of 3,989 sentences, of which only 94 were labeled as check-worthy – a staggering imbalance with only 2.36% of the dataset bearing the label of the target class. A few simple sentences from this training data, along with their speakers and labels, are presented in Table 1.

The test data was a collection of two political debates and five political speeches.[3] The total number of sentences in these two categories (*Debate* and *Speech*) were 2,815 and 2,064, respectively.

In this work, we did not employ any external knowledge other than domain-independent language resources such as parsers and lexicons. Instead, we focused extracting linguistic features indicative of check-worthiness.

[3] The lab task provided all seven files together, without this categorization into speeches and debates. We, however, chose to treat these differently since language use is very different in these two scenarios: *debates* consist of the interactive statements made by the candidates and the moderator, while *speeches* only have a single speaker, and there is no two-sided conversational structure.

2.3 Evaluation Framework

The evaluation was done on the test data provided as part of the task. This data was released much later to the participants, with the gold standard labels for the sentences in the test data withheld. Once we selected the models, we ran it on the entire test data, and used average precision to measure the quality of the output ranking. Average precision is defined as

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

where n_{chk} is the number of check-worthy sentences, n is the total number of sentences, $\text{Prec}(k)$ is the precision at cut-off k in the list of sentences ranked by check-worthiness, and $\delta(k)$ is the indicator function equaling 1 if the sentence at rank k is check-worthy, and 0 otherwise. The primary metric used by the Fact Checking Lab [24] for the check-worthiness task was mean average precision (MAP), defined simply as the mean of the average precisions over all queries.

3 Methodology

Our methodology is a hybrid of rule-based heuristics and supervised classification. The motivation for this approach was to test the extent to which check-worthiness can be determined based on language constructs without relying on encyclopedic knowledge. Moreover, our aim was to develop an approach that was not specific to the domain of politics. In this section, we describe the data processing, feature selection, and heuristics involved in building our classification models.

3.1 Data Processing

The first step of our processing involved normalizing the speaker names. We did this by adding speaker-specific rules in order to correctly match the speakers extracted from various sentences to the actual speakers associated with the sentences. For example, speakers in the test data included "Hillary Clinton (D-NY)", "Former Secretary of State, Presidential Candidate", and simply "Clinton". These are, of course, all referring to the same speaker.

Next, we noted that the training data consisted only of political debates where multiple entities (two political candidates, a moderator, and the occasional audience reaction) engage in a conversation. Due to the very nature of debates, the rhetorical structure is different from speeches delivered by a single speaker. The test data, however, also included political speeches. Therefore, we extracted all sentences attributed to a speaker to create sub-datasets. This formed a new training sample, which we then used to train models to identify check-worthy sentences from speeches[4]. To identify check-worthy sentences from political debates, we used the original training data to train the models.

[4] The provided training sample included two speeches, and both were by Donald Trump. As a result, for the purpose of this task, a single sub-dataset was created. The approach is independent of the speaker and the number of speakers, however.

Table 2. Constituent tags from the Penn Treebank.

Clause-Level	SBAR, SBARQ, SINV, SQ, S
Phrase-Level	ADJP, ADVP, CONJP, FRAG, INTJ, LST, NAC, NP, NX, PP, PRN, PRT, QP, RRC, UCP, VP, WHADJP, WHAVP, WHNP, WHPP, X

3.2 Feature Design and Selection

For both speeches and debates, we extracted a set of syntactic and semantic features to obtain a consistent knowledge representation, and converted every sentence into a vector in an abstract semantic space. The details of these features and the resultant feature vector are discussed below.

Sentence Embedding: Traditional supervised learning in natural language processing tasks have used vector spaces where dimensions correspond to words (or other linguistic units). This, however, is not in accordance with the well-known distributional hypothesis in linguistics: words that occur in similar contexts tend to have similar meanings [11]. This necessitates the representation of sentences in a low-dimensional semantic space where similar meanings are closer together. Modeling sentence meanings in a low-dimensional space is a topic of extensive research by itself, and beyond the scope of this work. Instead, we adopted a simple method that leverages word embeddings. We used the 300-dimensional pre-trained Google News word embeddings[5] to represent each word as a vector [23], and took the arithmetic mean of all such vectors corresponding to the words in a sentence to obtain an abstract sentence embedding.

Lexical Features: From the training data, we removed stopwords and stemmed the remaining terms using the Snowball stemmer [30].

Stylometric Features: Stylometry, the statistical analysis of variations in linguistic constructs, has been used with great success in distinguishing deceptive from truthful language [6,26], and objective from subjective remarks [19,21]. Accordingly, we surmised that capturing stylistic variation will aid in the identification of check-worthy sentences as well, especially since they are typically expected to appear factual and objective.

In order to obtain shallow syntactic features from each sentence, we extracted the part-of-speech (POS) tags, the total number of tokens, and the number of tokens in past, present, and future tenses. We were able to infer the tense from the POS tags (e.g., both VBD and VBZ are verb tags, but they indicate past and present tense, respectively). Additionally, we also extracted the number of negations in each sentence.

[5] Available at https://code.google.com/archive/p/word2vec/.

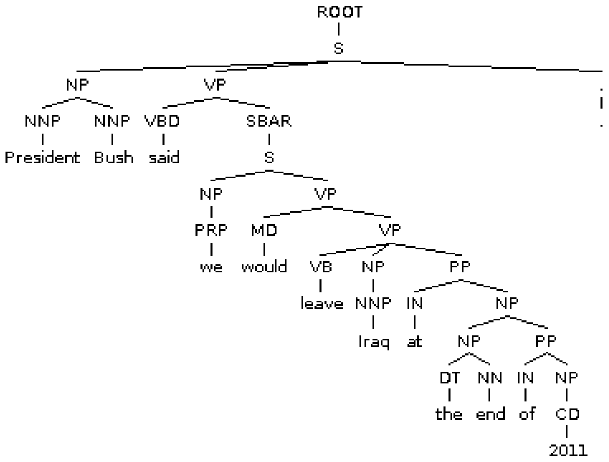

Fig. 1. The constituency parse tree of a check-worthy sentence from the training data: "President Bush said we would leave Iraq at the end of 2011." The size of the subtree under the subordinate clause (SBAR) is representative of the amount of information available provided about the action 'said' undertaken by the entity 'President Bush'.

More complex structural patterns of language, however, can only be captured by deep syntactic features. For that, we generated the constituency parse trees of all sentences, and selected clause-level and phrase-level tags. The number of words within the scope of each tag were included as the corresponding feature values. These tags, as defined in the Penn Treebank [2], are shown in Table 2. In addition to stylometry, the motivation behind using the number of words was to obtain a representation of the amount of information available under specific syntactic structures. Fig. 1 illustrates this point with the parse tree of a sentence from the training data that was labeled as check-worthy.

Semantic Features: We used the Stanford named entity recognizer (NER) [7] to extract the number of named entities in a sentence. Additionally, we appended an extra feature for named entities of the type PERSON.

Affective Features: We used the TextBlob [22] library to train a naïve Bayes classifier on the pioneering movie review corpus for sentiment analysis [27], and thereby obtained a sentiment score for each sentence. In addition to overt sentiment, we also used the connotation of words in a sentence as features. For this, we employed Connotation WordNet [16], which assigns a (positive or negative) connotation score to each word. For every sentence, we queried this lexicon and retrieved the connotation score of its words. Finally, the overall connotation of the sentence was attributed simply to the mean of these scores.

Additionally, we also utilized lexicons that contain information about the subjective or objective nature of words [35], whether they directly indicate or are typically associated with language that indicates bias [31], and whether they are typically used to voice positive or negative opinions [15]. For every sentence, we extracted the number of words in these categories (as defined by their scores in these lexicons), thus forming four new features: (i) subjectivity, (ii) direct bias, (iii) associated bias, and (iv) opinion.

Metadata Features: In addition to the syntactic and semantic features described above, we also included three binary non-linguistic features extracted from the training sample, indicating whether or not (i) the speaker's opponent is mentioned, (ii) the speaker is the anchor/moderator, or (iii) the sentence is immediately followed by intense reaction. The third feature is encoded in the training data as a 'system' reaction, as shown by the last sentence in Table 1.

Discourse Features: All the above features were extracted without regards to the category (i.e., *Debate* and *Speech*). Since debates involve an interactive discourse structure where sentences are often formed as an immediate response to statements made by others, we include **segments** from the debates. We adopt the approach taken by Gencheva et al [9] and regard a "segment" to be the maximal set of consecutive sentences by the same speaker. As features, we include the relative position of a sentence within its segment, and the number of sentences in the previous, current and subsequent segments.

Feature Selection The feature extraction processes described above yielded a very high-dimensional feature space. High-dimensional spaces, however, quickly lead to a decrease in the predictive power of models [32]. Moreover, given the extreme class imbalance, classification in such a space is likely to ignore important features indicative of the minority class (in this case, the 'check-worthy' sentences).

To reduce the dimensionality, we applied a feature selection module using the `scikit-learn` library [29]. As the first step, univariate feature selection was performed, and the 2,000 best features were selected based on χ^2-test. Next, armed with the observation that linear predictive models with L1 loss yield sparse solutions and encourage the vanishing coefficients for weakly correlated features [25], we used a support vector machine (SVM) model with linear kernel and L1 regularization to further remove the relatively unimportant features. This step was first done on the entire training data, and then combined with repeated undersampling (without replacement) for the majority class. Each iteration of this undersampling process resulted in a small but balanced training sample. A L1-regularized SVM learner was trained on every sample generated in this manner, and features with vanishing coefficients were discarded. The cumulative effect of these feature selection steps was a reduction of the feature space to 2,655 and 2,404 dimensions for identification of check-worthy claims from debates and speeches, respectively.

3.3 Heuristics

Certain heuristics were introduced to override the scores assigned by the classification models. These rules differed slightly based on (i) the category, i.e., speech or debate, and (ii) whether or not the 'strict' heuristics were deployed.

Algorithm 1 Heuristics for assigning the check-worthiness score $w(\cdot)$ to sentences.

```
Require: category $\in$ {SPEECH, 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
```

The strictness flag was introduced to control the threshold sentence size. When active, it would tend to discard more sentences.

These rules are specified in Algorithm 1. One particular rule required the identification of subjects in a sentence. To extract this information, we generated dependency parse trees of the sentences and counted the number of times any of the following dependency labels appeared: `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 Models

Our experiments comprised two supervised learning algorithms: support vector machines (SVM) and multilayer perceptrons (MLP). Additionally, we also built an ensemble model combing the two. In this section, we provide a description of these three models, along with their training processes.

For reasons described in Sec. 3.2, the SVM utilized a linear kernel with L1 regularization for feature selection. However, due to the propensity of the L1 loss function to miss optimal solutions, we used L2 loss in building the final model after completing feature selection. Our second model was the MLP. Here, we used two hidden layers with 100 units and 8 units in them, respectively. We used the hyperbolic tangent (tanh) as our activation function since it achieved better results when compared to rectified linear units (ReLU). Stochastic optimization was done with Adam [17]. To avoid overfitting, we used L2-regularization in both

Table 3. Results for the Check-Worthiness task of our submitted models: **MLP*** was the primary submission, along with two contrastive runs, MLP$_{\text{str}}$ and ENS (MLP with strict heuristics and the ensemble model, respectively). MLP$_{\text{none}}$ shows the results of the MLP without any heuristics being applied. The primary evaluation metric was mean avg. precision (MAP). The mean reciprocal rank (MRR), mean R-precision (MRP), and mean precision at k (MP@k) are also shown.

	MAP	**MRR**	**MRP**	**MP@1**	**MP@3**	**MP@5**	**MP@10**	**MP@20**	**MP@50**
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
ENS	0.1317	0.4139	**0.1523**	0.2857	0.1905	0.1714	**0.1571**	0.1571	**0.1429**
MLP$_{\text{none}}$	0.1086	0.4767	0.1037	0.2857	**0.2857**	0.2000	0.1286	0.1071	0.1000

SVM and MLP. Third, we built an ensemble model that combines SVM and MLP (without the strict heuristics). In this model, the final output score was a normalization (by standard deviation) of the results of SVM and MLP, and then computing the average.

For all three models, class imbalance was a hindrance during the training process. To overcome that, we used ADASYN[14], an adaptive synthetic sampling algorithm for imbalanced learning. For model selection, we used 3-fold cross-validation for debates, using two files for training and the remaining one for testing, to evaluate model performances and tune parameters. For speeches, we split the training sample into two halves (one file in each) for 2-fold cross-validation. The evaluation script was provided by the task organizers, with the mean average precision (MAP) being the primary evaluation metric.

MLP without the strict heuristics demonstrated the best results during the training process, so this was submitted for the primary run. For the two contrastive runs, we submitted (i) MLP with strict heuristics, and (ii) the ensemble model without the strict heuristics.

5 Results and Analysis

5.1 Empirical Results

The detailed performance of all three submissions we made is shown in Table 3. Even though MLP yielded the best training results without the strict heuristics, MLP$_{\text{str}}$ performed demonstrably better across multiple metrics on the test data. Our third model, the ensemble classifier, performed poorly in general compared to both MLP models. It did, however, achieve slightly better mean R-precision and mean precision at higher cutoffs ($k = 10$ and 50).

Without the inclusion of any heuristics, the performance of MLP dropped significantly. This was expected, since the heuristics were designed to address the flaws of the classifiers. This model was not among the submissions, but we include it here for comparison. The difference between MLP and MLP$_{\text{none}}$ quantifies the extent to which the rules help the supervised learners.

Table 4. Results from the primary submissions from all participants. We participated under the name *Prise de Fer*. The best results for each metric is shown in bold.

TEAM	MAP	MRR	MRP	MP@1	MP@3	MP@5	MP@10	MP@20	MP@50
Prise de Fer*	**0.1332**	**0.4965**	**0.1352**	**0.4286**	**0.2857**	0.2000	0.1429	**0.1571**	0.1200
Copenhagen	0.1152	0.3159	0.1100	0.1429	0.1429	0.1143	0.1286	0.1286	**0.1257**
UPV-INAOE	0.1130	0.4615	0.1315	0.2857	0.2381	**0.3143**	**0.2286**	0.1214	0.0866
bigIR	0.1120	0.2621	0.1165	0.0000	0.1429	0.1143	0.1143	0.1000	0.1114
fragarach	0.0812	0.4477	0.1217	0.2857	0.1905	0.2000	0.1571	0.1071	0.0743
blue	0.0801	0.2459	0.0576	0.1429	0.0952	0.0571	0.0571	0.0857	0.0600
RNCC	0.0632	0.3755	0.0639	0.2857	0.1429	0.1143	0.0571	0.0571	0.0486

Next, in Table 4, we present the comparison between the results obtained by all participants. This comparison was done only on the primary submission from each team. Our MLP model without the strict heuristics achieved the best MAP, MRR, and MRP scores. Further, it also outperformed the others in terms of correctly placing the check-worthy sentences at the very top of the ranked output list, as demonstrated by the mean precision at low values ($k = 1$ and 3).

5.2 Qualitative Analysis

Identifying check-worthy sentences is a difficult and novel task, and even the best model suffered from misclassification errors. Upon analyzing such mistakes made by the MLP models, we were able to discern a few reasons.

First, tense plays a logical role in check-worthiness, since future actions cannot be verified. However, the part-of-speech tagging often confuses the future tense with the present continuous (e.g., "We're cutting taxes."). Second, we observed that anecdotal stories are often highly prioritized as check-worthy, while they are not. These sentences are usually complex, with a lot of content, which makes it easy for the model to conflate them with other complex sentences pertaining to real events deemed check-worthy. Third, the presence of duplicate sentences in the data means that a misclassification gets amplified, while the presence of very similar sentences with different labels likely makes the feature selection stage discard potentially useful features.

At a more abstract level, rhetorical figures of speech play a critical role. They often break the structures associated with standard sentence formation. Several sentences that were misclassified exhibited constructs such as *scesis onomaton*, where words or phrases with nearly equivalent meaning are repeated. We conjecture that this makes the model falsely believe that there is more informational content in the sentence. Such figures of speech become even harder to handle when they occur across multiple speakers in debates. The conversational aspect of debates also causes another problem: quite a few sentences are short, and in isolation, would perhaps not be check-worthy. However, as a response to things mentioned earlier in the debate, they are.

Another complex issue leading to misclassification is the use of sentence fragments. This is sparingly used for dramatic effect in literature, but was seen with alarming frequency in the political debates due to the prevalence of ill-formed or partly-formed sentences stopping and then giving way to another sentence. In some cases, the fragments are portions of the sentence that the speaker repeats. An example of such a fragment is the sentence "Ambassador Stevens – Ambassador Stevens sent 600 requests for help.", where the phrase "Ambassador Stevens" is repeated.

A proper approach to deal with these hurdles is a complex matter in and by itself. We believe that our features are better suited for written language than speech or debate transcripts. In the presence of significantly more labeled data for check-worthiness, ablation studies that remove such sentences could provide empirical evidence of this intuition.

6 Conclusion and Future Work

We developed a hybrid system that combines a few rules with supervised learning to detect check-worthy sentences in political debates and speeches. To tackle the severity of class imbalance, our development also included a sophisticated feature selection process and special sampling methods. Our primary model achieved the best results among all participants over multiple performance metrics.

This work opens up several intriguing possibilities for future research in the field of fact-checking. First, we intend to study in greater details the linguistic forms of *informational* content. Shallow syntax has been explored to understand this aspect of language in sociolinguistics, and some work has even looked into deep syntactic features. This approach has, however, not yet been applied to identifying check-worthy sentences. Furthermore, more complex neural network structures need to be thoroughly investigated. Along this line, we will be investigating deep learning models with feedback control. A stringent and focused work on these issues will empower journalists and citizens alike to be better informed and more cognizant of false claims permeating news and social media now. To that end, we also need complementary advances in related areas like natural language querying, crowdsourcing, source identification, and social network analysis.

Acknowledgment: This work was supported in part by the U.S. National Science Foundation (NSF) under the award SES-1834597.

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. Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, CEUR-WS.org, Avignon, France (September 2018)

2. Bies, A., Ferguson, M., Katz, K., MacIntyre, R., Tredinnick, V., Kim, G., Marcinkiewicz, M.A., Schasberger, B.: Bracketing Guidelines for Treebank II Style Penn Treebank Project. University of Pennsylvania **97**, 100 (1995)
3. 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 Publishing Inc. (2012)
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. Feng, S., Banerjee, R., Choi, Y.: Syntactic Stylometry for Deception Detection. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2. pp. 171–175. Association for Computational Linguistics (2012)
7. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics. pp. 363–370. Association for Computational Linguistics (2005)
8. Flew, T., Spurgeon, C., Daniel, A., Swift, A.: The promise of computational journalism. Journalism Practice **6**(2), 157–171 (2012)
9. Gencheva, P., Nakov, P., Màrquez, L., Barrón-Cedeño, A., Koychev, I.: A context-aware approach for detecting worth-checking claims in political debates. In: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017. pp. 267–276 (2017)
10. Goode, L.: Social news, citizen journalism and democracy. New media & society **11**(8), 1287–1305 (2009)
11. Harris, Z.S.: Distributional Structure. Word **10**(2-3), 146–162 (1954)
12. Hassan, N., Li, C., Tremayne, M.: Detecting check-worthy factual claims in presidential debates. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management. pp. 1835–1838. CIKM (2015)
13. 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)
14. 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)
15. Hu, M., Liu, B.: Mining and Summarizing Customer Reviews. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 168–177. ACM (2004)
16. Kang, J.S., Feng, S., Akoglu, L., Choi, Y.: ConnotationWordNet: Learning Connotation over the Word+Sense Network. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers). pp. 1544–1554. Association for Computational Linguistics (June 2014)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Klayman, J.: Varieties of Confirmation Bias. In: Psychology of learning and motivation, vol. 32, pp. 385–418. Elsevier (1995)

19. Lamb, A., Paul, M.J., Dredze, M.: Separating Fact from Fear: Tracking Flu Infections on Twitter. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 789–795 (2013)
20. Le, D.T., Vu, N.T., Blessing, A.: Towards a text analysis system for political debates. In: Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities. pp. 134–139 (2016)
21. Lex, E., Juffinger, A., Granitzer, M.: Objectivity Classification in Online Media. In: Proceedings of the 21st ACM Conference on Hypertext and Hypermedia. pp. 293–294. ACM (2010)
22. Loria, S.: TextBlob: Simplified Text Processing. http://textblob.readthedocs.org/en/dev/ (2014)
23. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781 (2013)
24. 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)
25. Ng, A.Y.: Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In: Proceedings of the twenty-first international conference on Machine learning. p. 78. ACM (2004)
26. Ott, M., Choi, Y., Cardie, C., Hancock, J.T.: Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – Vol. 1. pp. 309–319. Association for Computational Linguistics (2011)
27. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. pp. 79–86. Association for Computational Linguistics (2002)
28. Patwari, A., Goldwasser, D., Bagchi, S.: TATHYA: A Multi-Classifier System for Detecting Check-Worthy Statements in Political Debates. In: Proceedings of the 26th ACM International Conference on Information and Knowledge Management. pp. 1–4. CIKM (2017)
29. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
30. Porter, M.F.: Snowball: A Language for Stemming Algorithms. http://snowball.tartarus.org/texts/introduction.html (2001)
31. Recasens, M., Danescu-Niculescu-Mizil, C., Jurafsky, D.: Linguistic Models for Analyzing and Detecting Biased Language. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 1650–1659 (2013)
32. Trunk, G.V.: A Problem of Dimensionality: A Simple Example. IEEE Transactions on Pattern Analysis and Machine Intelligence **1**(3), 306–307 (1979)
33. 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)
34. Werner, J.S., Tankard Jr, J.W.: Communication theories: Origins, methods and uses in the mass media (1992)

35. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. pp. 347–354. Association for Computational Linguistics (2005)
36. 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)