TATHYA: A Multi-Classifier System for Detecting Check-Worthy Statements in Political Debates

ABSTRACT
Fact-checking political discussions has become an essential cog in computational journalism. This task encompasses an important sub-task—identifying the set of statements with ‘check-worthy’ claims. Previous work has treated this as a simple text classification problem discounting the nuances involved in determining what makes statements check-worthy. We introduce a dataset of political debates from the 2016 US Presidential election campaign annotated using all major fact-checking media outlets. We study the characteristics of check-worthy statements and show that there is a need to model conversation context, debate dynamics and implicit world knowledge. We design a multi-classifier system TATHYA 1, that models latent groupings in data and improves state-of-art systems by 19.5% in F1-score on a held-out test set, gaining primarily gaining in Recall.

KEYWORDS
computational journalism, natural language processing, clustering

1 INTRODUCTION
Social media is widely used by politicians, especially during election campaigns, to promote their message and often, bias public opinion in their favor on important issues. The statements made often have multiple interpretations amongst the public leading to fake news [1]. To tackle this, there has been industry an wide effort in journalism towards fact-checking. Claims made during the 2016 US presidential debates were actively scrutinized by multiple fact-checking organizations in real-time. 2

Today, fact-checking efforts are primarily manual, lacking in coverage and consensus across different outlets. There are also constraints on budget and man-power. Studies in political journalism [6] have suggested that the rise in fact-checking is motivated more by "professional motives within journalism" than audience satisfaction. With check-worthy statements automatically detected, the fact-checkers can focus on sieving through a reduced corpus, increasing coverage and quality of their output.

Past research has addressed the problem of detecting checkworthiness [8] and verification of simple numerical claims [17, 19]. In these works there is an inherent assumption that properties of the statement itself is sufficient for performing those tasks. We discuss a short excerpt from out dataset in Table 1 to show that this task is in-fact, much more nuanced. We can see that almost all statements have an associated claim that is checkable e.g., in statement (5) the claim could be the U.S. government is not innovating, but only statements (2), (3) and (8) were fact-checked. This suggests that ‘check-worthy’ is a subset of ‘checkable’ and detecting what is check-worthy becomes even harder. Furthermore, check-worthiness is not consistent across statements with similar content—same content may be check-worthy or not depending on hidden factors e.g., speaker and opposition stance on the matter or current world context. We model factors that affect fact-checking —

Table 1: An excerpt from our dataset showing nuances in fact-checking. Statements fact-checked (2,3,8) are italicized. Implied information is shown in blue. Pronouns that need to be resolved are marked red and corresponding resolution entities are marked green.

Excerpt: Carly Fiorina, 5th Republican Primary
1. Let me tell you a story.
2. Soon after 9/11, I got a phone call from the NSA.
3. I stopped a truckload of equipment (for NSA).
4. It was escorted by the NSA into headquarters.
5. We need the private sector’s help, because government is not innovating.
6. Technology is running ahead by leaps and bound.
7. The private sector will help, just as I helped after 9/11.
8. But they must be engaged (with NSA), and they must be asked.

2 RELATED WORK
Automating fact checking [8, 18] is so far limited to very specific domains that can leverage existing knowledge bases and numerical statements [10, 17, 19], or existing knowledge by the user [14]. In this work we focus on one aspect of this challenge, identifying what type of content should be checked. In addition to the inherent bias in deciding what should be checked, there are substantial linguistic
challenges in analyzing such statements successfully. Some of these challenges bear resemblance to existing work. For example, identifying the arguments and how they relate to one another [11, 15], the discussion strategies used by the speakers [16]. Identifying check-worthy claims could also be considered as distantly related to the deception detection task [12, 13], however current work on deception detection builds on general representations of deception and bias, expressed through word choice and syntactic patterns [5, 7], and do not address the challenges of fact checking, such as pragmatic inferences and latent knowledge representation.

3 DATASET
We create our dataset by gathering transcripts from primary debates (7 Republican and 8 Democratic) and presidential debates (3 presidential one vice-presidential) which form our development and held-out test set respectively. We also include Donald Trump’s Presidential Announcement Speech to our development set to analyze a discourse by only one speaker. For each of these transcripts, we split at granularity of a sentence, which forms the unit of check-worthy if any of the fact-checking organizations listed in Table 2 checked it. We don’t use in-house annotators to prevent likely opinion bias and also train our system on real fact-checker outputs. A total of 21,700 statements were collected with 1,085 of them marked check-worthy. Since some statements were very short, we removed those with less than 2 tokens (tokens are extracted after removing frequently occurring words and stop-words) from our dataset. After this, the corpus had 15,735 statements, out of which 967 are marked check-worthy (6.1% of the corpus). All our analysis is based only on the development set and we use the test set only for final evaluation.

<table>
<thead>
<tr>
<th>R</th>
<th>D</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>8781</td>
<td>6454</td>
</tr>
<tr>
<td>Check-worthy</td>
<td>290</td>
<td>318</td>
</tr>
<tr>
<td>Check-worthy: Organization Wise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Washington Post</td>
<td>67</td>
<td>152</td>
</tr>
<tr>
<td>factcheck.org</td>
<td>63</td>
<td>113</td>
</tr>
<tr>
<td>Politifact</td>
<td>72</td>
<td>37</td>
</tr>
<tr>
<td>PBS</td>
<td>35</td>
<td>47</td>
</tr>
<tr>
<td>CNN</td>
<td>29</td>
<td>33</td>
</tr>
<tr>
<td>NY Times</td>
<td>29</td>
<td>19</td>
</tr>
<tr>
<td>Fox News</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>USA Today</td>
<td>14</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 2: Data. R and D denote statements by Republicans and Democrats respectively.

check-worthy if any of the fact-checking organizations listed in Table 2 checked it. We don’t use in-house annotators to prevent likely opinion bias and also train our system on real fact-checker outputs. A total of 21,700 statements were collected with 1,085 of them marked check-worthy. Since some statements were very short, we removed those with less than 2 tokens (tokens are extracted after removing frequently occurring words and stop-words) from our dataset. After this, the corpus had 15,735 statements, out of which 967 are marked check-worthy (6.1% of the corpus). All our analysis is based only on the development set and we use the test set only for final evaluation.

4 EMPIRICAL ANALYSIS
Organizational Subjectivity: In our dataset there are inconsistencies amongst different organizations on how they fact-checked debates, e.g., Washington Post and NY Times checked 16 and 29 statements in the 5th Republican Primary Debate respectively, but their overlap is only 6 sentences. Also, Washington Post and factcheck.org have checked more statements by Democrats, whereas Politifact and NPR seem to have focused more on Republicans.

Party-Wise Differences: We extract the named entities in check-worthy statements for each party. It is interesting to find that Out of the 71 and 94 named entities mentioned by Republicans and Democrats respectively, there are only 21 in common, prominent ones being Americans, ISIS, Bush, Clinton, Donald Trump, Obama, White House, Social Security and NSA. Majority of the entities are specific to the party (70.4% for Republicans and 77.7% for Democrats). This shows that entities in conversation context e.g., during a Republican Primary, and the topic of discussion might be helpful in determining check-worthiness.

Human Evaluation: Our dataset is highly unbalanced with 6.1% statements marked check-worthy. To understand complexity of this task, we ask two human annotators – a graduate and an undergraduate student – with explicit information on the gold labels for a sample of 1177 (~10%) sentences from our development set – to find similar, check-worthy statements. They marked another 145 statements as check-worthy (considering only those on which the annotators agree). This shows that there is latent information that governs whether statements with very similar content would be check-worthy.

5 MULTI-CLASSIFIER SYSTEM
Multi-classifier systems have been shown to improve performance in cases where a single classifier system lacks expressiveness for the task at hand [4]. We essentially want to learn a latent grouping of our dataset that best describes the target output function, in our case, given a statement, whether it is check-worthy or not. Such latent representations have been shown to improve performance in the past [3]. To achieve this we design a classifier system as shown in algorithms 1 and 2. In the training step 1, we first cluster the training data into $k$ groups which we use as initialization seeds for the algorithm. In steps 2-7 we learn the best groupings of our data $g_1, ... g_k$ which allows us decrease ambiguity of classification and improve training performance by learning separate classifiers $C_1, ... C_k$. Prediction is done simply using the most-confident classifier for each sample.

6 FEATURE DESIGN
Here describe the different feature classes that we use and the design of multi-layer classifier.

Topics of Discussion: Claims in certain topics, like foreign policy, health-care, gun control etc. are more likely to be checked by fact-checkers. We train an LDA topic model [2] on transcripts from all presidential debates (from 1976 to 2016) and tune the number of topics to 20. We and generate a topic probability distribution

---

5Statements from moderators are also included. 13 statements out of 1239 were fact-checked.
6For presidential debates we collected labels from only NPR to ensure no overlap of organizations between development and test set.
We restrict propagation to entities of types normalization by propagating chained named-entities along a dis-ond and third person pronouns in a discussion. We perform text tokens appearing in more than 20% also used, e.g., 'Affordable Care Act'. Stop words are removed and

For each statement the count of each pos-tag are also used.

Using

\[ P = \frac{\text{correct}}{\text{predicted}}, \quad R = \frac{\text{correct}}{\text{gold}} \quad \text{and} \quad F = 2 \cdot \frac{P \cdot R}{P + R}. \]

We used Stanford CoreNLP\(^{10}\) and NLTK\(^{10}\) for tokenization, POS-tagging, NER-tagging and Coreference-Resolution. We train LDA topic model using Gibbs sampling\(^{11}\). We use linear SVM classifiers trained using scikit-learn\(^{12}\). We tune hyper-parameters of the system using grid-search on cross-validation. For SVM we keep penalty parameter \( C = 0.1 \) and class_weight proportional to half of class-ratio for best cross-validation performance. We use Kmeans to cluster data in our multi-classifier system.

8 RESULTS

8.1 Ablation Study

We describe here the model performance for the combination of different feature sets described in the last section for a single classifier. We divide our development set into 16 folds – each fold contains statements from one debate/speech – and perform 16-fold cross-validation by training on all-but-one and testing on the remaining debate. The results are presented in Table 3. We can see that a

<table>
<thead>
<tr>
<th></th>
<th>( P )</th>
<th>( R )</th>
<th>( F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClaimBuster</td>
<td>0.194</td>
<td>0.32</td>
<td>0.241</td>
</tr>
<tr>
<td>( \text{bow} )</td>
<td>0.194</td>
<td>0.337</td>
<td>0.241</td>
</tr>
<tr>
<td>( \text{bow, pos} )</td>
<td>0.181</td>
<td>0.399</td>
<td>0.245</td>
</tr>
<tr>
<td>( \text{bow, pos, ent} )</td>
<td>0.185</td>
<td>0.411</td>
<td>0.251</td>
</tr>
<tr>
<td>( \text{bow, pos, ent, pos-T}, t_1, t_2 )</td>
<td>0.193</td>
<td>0.435</td>
<td>0.263</td>
</tr>
</tbody>
</table>

Table 3: Cross-validation performance of detecting check-worthy claims for development. \( \text{bow} \) is bag-of-words, \( \text{pos} \) is pos-tag counts, \( \text{ent} \) is entity-type counts, \( \text{pos-T} \) is POS-tuples, \( t_1 \) is topic agreement, \( t_2 \) is entity history. Text normalization is used for before feature extraction.

simple bag-of-words model on normalized text performs as good as ClaimBuster. Adding pos-tags and entity-types improves the model by 4%. Adding pos-tuples, topics and entity history improves the Fscore to 0.263 with primarily gain in recall over ClaimBuster. We call this system TATHYA-SVM.

8.2 Multi-Classifier Performance

We evaluate our multi-classifier system by first training and predicting on the training set for various values of \( K \in [2, 6] \). Beyond 7, some of the initial clusters had no +ve samples. We follow the algorithm described in Algorithm 1 and compute training F1-score after each iteration. The results are shown in Fig. 1 (a).

The training accuracy increases with the iterations and after a point it saturates. We find that generally with higher \( K \) training

\(^{8}\)https://stanfordnlp.github.io/CoreNLP/

\(^{10}\)http://www.nltk.org/

\(^{11}\)https://pypi.python.org/pypi/lda

problem is made difficult by a confluence of factors. Acknowledging the difficulties, we design a classifier system that uses features to model these factors and also attempts to learn latent groupings of data. Comparing our system TATHYA to the current state-of-the-art, ClaimBuster, on the presidential debates, we find an improvement of 19.5% in F1-score and 67% in recall. In future work, we will attempt to learn better latent representations that would enable to increase the expressiveness of the classifier and further improve performance.

REFERENCES


Table 4: Performance comparison on held-out test set of presidential debates from US Presidential Elections 2016.

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClaimBuster</td>
<td>0.226</td>
<td>0.148</td>
<td>0.179</td>
</tr>
<tr>
<td>TATHYA-SVM</td>
<td>0.227</td>
<td>0.194</td>
<td>0.209</td>
</tr>
<tr>
<td>TATHYA-MULT</td>
<td>0.188</td>
<td>0.248</td>
<td>0.214</td>
</tr>
</tbody>
</table>

Figure 1: Performance of the multi-classifier system on training (a) and test (b) set respectively.