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

ABSTRACT

Fact-checking political discussions has become an essential clog 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

ACM Reference format:
. 2017. TATHYA: A Multi-Classifier System for Detecting Check-Worthy Statements in Political Debates. In *Proceedings of ACM Conference on Information and Knowledge Management, Marina Bay, Singapore, November 2017 (CIKM'17),* 4 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

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]. [3]

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 check-worthiness [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 –

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.*

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.

debate context, important topics of discussion and the nature of claims and to identify latent groupings of data that causes ambiguity within similar samples, we design multi-classifier system that learns the best latent representation which describing the output. Our system TATHYA outperforms the current state-of-the-art ClaimBuster [4] [8, 9] by 19.5% in F1-score on a held-out-set of 4 presidential debates.

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

[1] TATHYA means 'fact' in Hindi, signifying our efforts to automate fact-checking.
[2] Bill Adair. 2016. What Happened on Election Day. (2016). https://tinyurl.com/ l9kwxqj
[3] Tara Golshan. 2016. The importance of fact-checking the debate in real time, according to an expert. (2016). http://www.vox.com/2016/9/26/13063004

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

CIKM'17, November 2017, Marina Bay, Singapore

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

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 checking similar to previous work [8, 19]. A statement is labeled as check-worthy if any of the fact-checking organizations listed in Table 2 checked it [6]. 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.

	R	D	Total
Primary Debates			
All	8781	6454	15235
Check-worthy	290	318	608
Check-worthy: Organization Wise			
Washington Post	67	152	219
factcheck.org	63	113	176
Politifact	72	37	109
PBS	35	47	82
CNN	29	33	62
NY Times	29	19	48
Fox News	13	16	29
USA Today	14	9	23
Presidential Debates [5]			
All	2956	2270	6465
Check-worthy (NPR)	300	164	477

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

[5]Statements from moderators are also included. 13 statements out of 1239 were fact-checked.

[6]For presidential debates we collected labels from only NPR to ensure no overlap of organizations between development and test set

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 $g_1, \ldots g_k$ which allows us decrease ambiguity of classification and improve training performance by learning separate classifiers $C_1, \ldots C_k$. 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[7] (from 1976 to 2016) and tune the number of topics to 20. We and generate a topic probability distribution

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

Algorithm 1: Multi-classifier System Training

Input : input samples X, input labels Y, num_groups k, max_iter, tol

Output : classifiers $C_1, \ldots C_k$

```
Cluster data into k groups g_1, ...g_k
for iter ≤ max_iter do
    Train classifier C_i on group g_k for i ∈ [1, k]
    Predict using C_i on X for i ∈ [1, k]
    for each (x, y) ∈ (X, Y) do
        Assign x to group g* where C* has highest confidence
          conf for the correct output label y.
    end
end
```

Algorithm 2: Multi-classifier System Prediction

Input : input samples X, classifiers $C_1, \ldots C_k$

Output : output labels Y

```
for each x ∈ X do
    Predict y_i using C_i for i ∈ [1, k]
    Using C* having highest confidence, predict label y*
end
```

for each statement using the trained model. Then we define a context size (say c_1), and for $c\_1/2$ previous statements and $c\_1/2$ following statements we compute the cosine similarity.

Entity History: For each statement we create a entity history of size h, which has all entities appearing in h previous statements of that debate. For any entity in history that is repeated in the current statement, if the speaker is same we activate a feature (*entity_type*, *discuss*) else (*entity_type*, *repeat*). We also keep the counts of all *entity_type*s for each statement as features.

Part-of-Speech tuples: Claims often have a dependency structure (*subj*,*verb*,*obj*). We want to target *subj* and the *verb* and capture the sense (+ve or −ve) of self and opponent references. To achieve this we define POS target tuples [8]

- *(noun_tag, verb_tag) e.g.,* 'Sanders has'
- *(noun_tag, verb_tag, neg) e.g.,* 'She did not'
- *(noun_tag, neg, verb_tag) e.g.,* 'I never told'

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

Bag-Of-Words: We use bag-of-unigrams using tf-idf weighting as a baseline model. Very frequently occurring n-grams (phrases) are also used, *e.g.,* 'Affordable Care Act'. Stop words are removed and tokens appearing in more than 20% of the sentences are removed. We also include sentiment class ($+ve$ or $-ve$) and number of tokens for each statement as features.

Text Normalization: It is common to refer to entities using second and third person pronouns in a discussion. We perform text normalization by propagating chained named-entities along a discussion. We restrict propagation to entities of types *person*, *org* and *misc*. We exclude resolution of *we*, since it is particularly confusing for an automated system and increases error in normalization. The normalized text is then used for extracting features.

7 METRICS AND EXPERIMENTAL SETUP

In all our evaluation we use $Precision(P)$, $Recall(R)$ and $F1score(F)$ for the check-worthy class as our metrics of comparison. where $P = \frac{\#correct}{\#predicted}$, $P = \frac{\#correct}{\#gold}$ and $F = 2\frac{P \times R}{P+R}$. We used Stanford CoreNLP[9] 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 simple bag-of-words model on normalized text performs as good as ClaimBuster. Adding pos-tags and enitity-types improves the model by 4%. Adding pos-tuples, topics and entity history improves the F1score to 0.263 with primarily gain in recall over ClaimBuster. We call this system Tathya-SVM.

	P	R	F
ClaimBuster	0.194	0.32	0.241
bow	**0.194**	0.337	0.241
bow, pos	0.181	0.399	0.245
bow, pos, ent	0.185	0.411	0.251
bow, pos, ent, *pos-T*, t_1, t_2	0.193	**0.435**	**0.263**

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

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

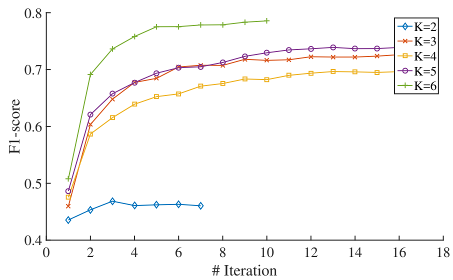
[8] where $noun\_tag$ ∈ {'NN', 'NNS', 'NNP', 'NNPS', 'PRP', 'PRP', 'WP', 'WP'}, $verb\_tag$ ∈ {'VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ'} and neg ∈ {'neither', 'never', 'no', 'not', 'none'}

[9] https://stanfordnlp.github.io/CoreNLP/

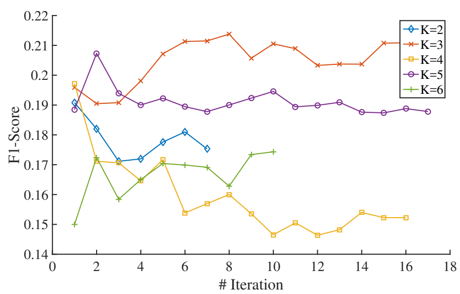
[10] http://www.nltk.org/

[11] https://pypi.python.org/pypi/lda

[12] http://scikit-learn.org/stable/modules/svm.html



(a) Train (Baseline with Single SVM = 0.41)



(b) Test (Baseline with Single SVM = 0.209)

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

score saturates faster. After 8 iterations the differences F1-score are negligible in all cases (< 0.05). The performance on the test set is shown in Fig. 1 (b). We find that only for $K = 3, 5$ there is consistent performance. We conclude that for $K = 3$ the latent groupings are optimal for our classification task, in the sense that they best describe our final output function. We find the best F1-score of 0.214 for $K = 3$, an improvement of 2.4% over TATHYA-SVM along with a 28.8% improvement in recall. We call this system TATHYA-MULT.

	P	R	F
ClaimBuster	0.226	0.148	0.179
TATHYA-SVM	**0.227**	0.194	0.209
TATHYA-MULT	0.188	**0.248**	**0.214**

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

8.3 Comparison With ClaimBuster

For a fair comparison with ClaimBuster we use the test set comprising of only the presidential and vice-presidential debates. To compute the output for ClaimBuster we use their web-api which provides a score in $[0, 1]$ for a given statement. We classify a statement as check-worthy if the score is ≥ 0.5; this is the threshold used by authors in the paper[8]. Both our models out-perform ClaimBuster on the test set by 16.8% and 19.5% respectively in F1-score.

9 CONCLUSION AND FUTURE WORK

In this paper, we tackle the problem of detecting whether statements made by politicians are check-worthy or not. We find that this 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

[1] Hunt Allcott and Matthew Gentzkow. 2017. *Social media and fake news in the 2016 election.* Technical Report. National Bureau of Economic Research.

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

[3] Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative learning over constrained latent representations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics, 429–437.

[4] Y-Y Chou and Linda G Shapiro. 2003. A hierarchical multiple classifier learning algorithm. *Pattern Analysis & Applications* 6, 2 (2003), 150–168.

[5] Song Feng, Ritwik Banerjee, and Yejin Choi. 2012. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2.* Association for Computational Linguistics, 171–175.

[6] Lucas Graves, Brendan Nyhan, and Jason Reifler. 2016. Understanding Innovations in Journalistic Practice: A Field Experiment Examining Motivations for Fact-Checking. *Journal of Communication* 66, 1 (2016), 102–138.

[7] Stephan Greene and Philip Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics.* Association for Computational Linguistics, 503–511.

[8] Naeemul Hassan, Bill Adair, James T Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. 2015. The quest to automate fact-checking. *Computation and Journalism Symposium* (2015).

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

[10] Julien Leblay. 2017. A Declarative Approach to Data-Driven Fact Checking. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.* 147–153.

[11] Marco Lippi and Paolo Torroni. 2015. Context-Independent Claim Detection for Argument Mining.. In *IJCAI*, Vol. 15. 185–191.

[12] Rada Mihalcea and Carlo Strapparava. 2009. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers.* Association for Computational Linguistics, 309–312.

[13] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. 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-Volume 1.* Association for Computational Linguistics, 309–319.

[14] Jeff Pasternack and Dan Roth. 2010. Knowing what to believe (when you already know something). In *Proceedings of the 23rd International Conference on Computational Linguistics.* Association for Computational Linguistics, 877–885.

[15] Isaac Persing and Vincent Ng. 2016. End-to-end argumentation mining in student essays. In *Proceedings of NAACL-HLT.* 1384–1394.

[16] Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International Conference on World Wide Web.* International World Wide Web Conferences Steering Committee, 613–624.

[17] James Thorne and Andreas Vlachos. 2017. An Extensible Framework for Verification of Numerical Claims. *EACL 2017* (2017), 37.

[18] Andreas Vlachos and Sebastian Riedel. 2014. Fact Checking: Task definition and dataset construction. *ACL 2014* (2014), 18.

[19] Andreas Vlachos and Sebastian Riedel. 2015. Identification and verification of simple claims about statistical properties. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, 2596–2601.