ClaimPortal: Integrated Monitoring, Searching, Checking, and Analytics of Factual Claims on Twitter

Sarthak Majithia **Fatma Arslan** **Sumeet Lubal**
Damian Jimenez **Priyank Arora** **Josue Caraballo** **Chengkai Li**
Department of Computer Science and Engineering
The University of Texas at Arlington

Abstract

We present ClaimPortal, a web-based platform for monitoring, searching, checking, and analytics of factual claims on Twitter. We explain the architecture of ClaimPortal, its components and functions, and the user interface. While the last several years have witnessed a substantial growth in interests and efforts in the area of computational fact-checking, ClaimPortal is a novel infrastructure in that fact-checkers have largely skipped factual claims in tweets. It can be a highly powerful tool to both general web users and fact-checkers. It will also be an educational resource in helping cultivate a society that is less susceptible to falsehoods.

1 Introduction

Our society is struggling with an unprecedented amount of falsehood that can do harm to wealth, health, democracy, and national security. In domestic political discourses, politicians repeat false claims even after they are debunked. “Fake news” is fabricated to spread derogatory rumors, promote societal and political tensions, manipulate public opinion, and even influence an election outcome. We have seen how misinformation can have an impact in our world, most poignantly through how an election, a cornerstone of our democracy, can be affected.

At news organizations such as The Washington Post, New York Times and FactCheck.org, professional fact-checkers take on the hard battle to counter misinformation and disinformation. They vet claims by analyzing relevant data and documents and publishing their verdicts. For instance, PolitiFact.com gives factual claims truthfulness ratings such as true, half true, false, and even “pants on fire”. However, there is simply far more misinformation on the web than what fact-checkers can keep up with. The process of fact-checking is laborious and intellectually demanding, as it takes the professionals about one day to research and write a typical article about a factual claim (Hassan et al., 2015a). This difficulty leaves many harmful claims unchecked, since fact-checking organizations can only use their limited resources to focus on national events and prominent figures.

This problem of unchecked claims is exacerbated on social media. On the one hand, fact-checkers have largely skipped social media in their fact-checking activities, due to limited resources. On the other hand, a large number of false claims, likely much more than those in traditional media, are being spread through social media. This can be due to the compounded effect of several factors: social media platforms have become increasingly important to public figures and organizations in engaging with voters and citizens; mobile devices have brought an age in which sharing and disseminating information is easy for anyone, including both malicious and unintentional creators of falsehoods; the falsehoods are further replicated and amplified by social media bots and clickbait articles. The consequence can be devastating. For instance, a recent study reports that a sample of 140,000 Twitter users in the battleground state of Michigan shared as many junk news items as professional news during the final ten days of the 2016 election, each constituting 23% of the web links they shared on Twitter in that period. [1]

In this paper we present ClaimPortal, a web-based platform for monitoring, searching, checking, and analytics of factual claims on Twitter. ClaimPortal is available at `https://idir.uta.edu/claimportal` and its demo video can be found at `https://vimeo.com/329947070`. ClaimPortal continuously collects tweets and

[1] `http://politicalbots.org/?p=1064`

monitors factual claims embedded in tweets. It is *integrated with fact-checking tools*, including a claim matcher which finds known fact-checks matching any given tweet, a claim spotter which scores each claim and the corresponding tweet based on their check-worthiness, i.e., how important it is to fact-check them. ClaimPortal provides an intuitive and convenient *faceted search interface* that assists its users to sift through these factual claims in tweets through filtering conditions on dates, twitter accounts, content, hashtags, check-worthiness scores, and types of claims. ClaimPortal also provides simple *analytics and visualization* tools for discovering patterns pertinent to how certain twitter accounts make claims, how different types of claims are distributed, and so on.

2 Related Work

While the initial call to arms to research on computational fact-checking was made nearly a decade ago (Cohen et al., 2011), the last several years have witnessed a substantial growth in interests and efforts in this arena. These efforts tackle various fronts, from detecting important factual claims that are worth checking (Hassan et al., 2015b; Jimenez and Li, 2018), to using databases for discerning factual claims' robustness (Wu et al., 2017) and truthfulness (Ciampaglia et al., 2015; Shi and Weninger, 2016; Jo et al., 2018), to building end-to-end fact-checking systems (Babakar and Moy, 2016; Hassan et al., 2017a,b). ClaimPortal is a novel infrastructure in that fact-checkers have largely skipped factual claims in tweets.

3 System Architecture and Components

3.1 System Architecture

ClaimPortal is composed of a front-end web based GUI, a MySQL database, an Elasticsearch[2] search engine, an API, and several decoupled batch data processing components (Figure 1). The system operates on two layers: the front-end presentation layer, and the back-end data collection and computation layer. The front-end allows users to narrow down search results by applying multiple filters. Keyword search on tweets is powered by Elasticsearch which is coupled with querying the database to provide additional filters. Additionally, it provides numerous visualized graphs. The back-end layer consists of the data gathering process, which includes pre-processing of tweets, computing check-worthiness scores of tweets using the public ClaimBuster API (Hassan et al., 2017a), Elasticsearch batch insertion, detecting claim types of tweets, and finding similar fact-checked claims for each tweet by employing ClaimBuster fact-checking API. ClaimPortal stays up-to-date with current tweets by periodically calling the Twitter REST API.

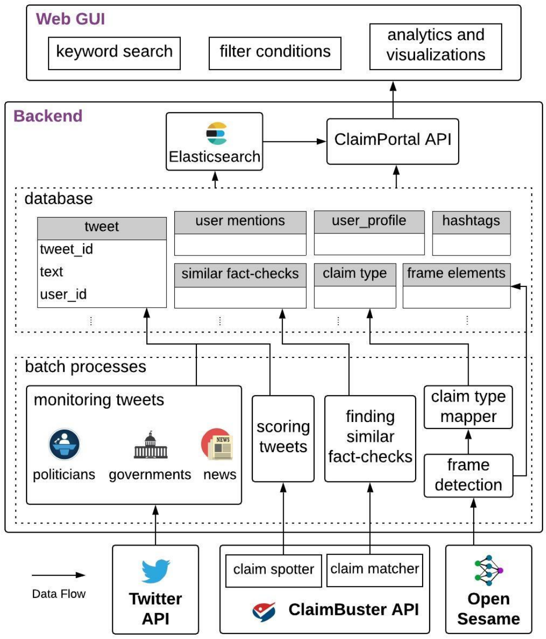

Figure 1: ClaimPortal system architecture.

3.2 Tweet Monitoring, Processing, and Storing

ClaimPortal at this moment focuses on politically-charged tweets, but will be expanded to eventually cover all types of tweets. We curated a list of prominent Tweet handles in U.S. politics that include but are not limited to house representatives and senators in the Congress, governors, city mayors, U.S. Cabinet members, other government officials, and political teams of news media. We then made use of the *user_timeline* endpoint of the Twitter REST API to navigate through each user's timeline and collected their tweets. More specifically, we navigated through the historic data of a user's timeline, which is a one-time process. We then keep our data up-to-date by continuously monitoring newly posted tweets. As of April 10, 2019, ClaimPortal monitors 3,200 Twitter handles and has collected approximately 3.3 million tweets after being deployed in mid-January 2019. We are

[2] https://www.elastic.co/products/elasticsearch

working on substantially expanding the curated list of Twitter handles.

ClaimPortal's back-end data collection and computation layer focuses on data processing and storage. The Twitter REST API provides us with the necessary data. However, the system does not require all of it. In fact, a lot of the API's response is discarded to keep our database size small and yet sufficient enough to provide all the information needed by the portal. This is achieved through the ClaimPortal API. ClaimPortal API is a web service designed using Python and the Flask [3] micro-framework. It provides API end points for loading tweets on the web based GUI, search for hash-tags, search for users to apply as from-user and user-mentions filters. Based on the keyword search and filters requested by a user, ClaimPortal API queries the database to find the resulting list of tweet ID's and returns it as a JSON response. A tweet ID is a unique number assigned to a tweet by Twitter. By using Twitter's card API [4] the system dynamically populates the latest activity of a tweet at the front-end, based on its ID.

The MySQL database has several normalized tables. For each tweet the database stores its text, when it was created, and who tweeted it. The database also stores information about re-tweets and quoted-tweets, hash-tags and URLs mentioned in the tweets, and information about the accounts mentioned in the tweets.

ClaimPortal uses Elasticsearch to support keyword search over the stored tweets. Since Elasticsearch is equipped with incremental indexing, the system periodically feeds Elasticsearch the delta tweets since last update for indexing. For this the system uses a decoupled background batch process that takes care of incrementally inserting tweets and uodating the Elasticsearch index.

3.3 Claim Spotter

In ClaimPortal, each tweet is given a check-worthiness score which denotes whether the tweet has a factual claim of which the truthfulness is important to the public. This score is obtained by probing the ClaimBuster API,[5] a well-known fact-checking tool, developed by our research group, that is being used by professional fact-checkers on a regular basis (Adair et al., 2019). ClaimBuster (Hassan et al., 2017a; Jimenez and Li, 2018) is a classification and ranking model trained on a human-labeled dataset of 8,000 sentences from past U.S. presidential debates. Human coders labeled each sentence into one of the following three categories:
(1) A **check-worthy factual sentence** contains a factual claim and the general public would be interested in knowing its truthfulness. E.g., "He voted against the first Gulf War." (2) An **unimportant factual sentence** has a factual claim that is not check-worthy. E.g., "Two days ago we ate lunch at a restaurant." (3) A **non-factual sentence** contains no factual claim. Opinions, declarations, and many questions fall under this category. E.g., "But I think it's time to talk about future."

Trained on the aforementioned dataset, the ClaimBuster API returns a check-worthiness score for any given text. The score is on a scale from 0 to 1, ranging from least check-worthy to most check-worthy. The background task of probing ClaimBuster API for getting scores for tweets is another batch process, in parallel with the tweet collection and the Elasticsearch indexing processes.

3.4 Detecting Claim Types

ClaimPortal uses tweets to gain insights into factual claims that are being spread, by whom, how often, and whether they are true. To answer these questions we categorize tweets by the types of factual claims they promote. We employed a collection of FrameNet frames (Baker et al., 1998) and created several new frames specifically for factual claims. We then adopted the study of mapping frames to event types (Spiliopoulou et al., 2017).

3.4.1 Frame detection

FrameNet is a linguistic resource for English comprised of 1,224 manually established semantic *frames*. Each frame provides information about both the linguistic and the semantic structure of a type of event, situation, object, or relation along with its participants. The participants, called *frame elements*, are frame-specific semantic roles that provide additional information. Each frame is evoked by a set of lexical units, or words, which are a composition of the lemma and meaning of the word.

We created new frames after conducting a survey of existing fact-checks from PolitiFact [6] and

[3] http://flask.pocoo.org
[4] https://developer.twitter.com/en/docs/tweets/optimize-with-cards
[5] https://idir.uta.edu/claimbuster/
[6] https://www.politifact.com

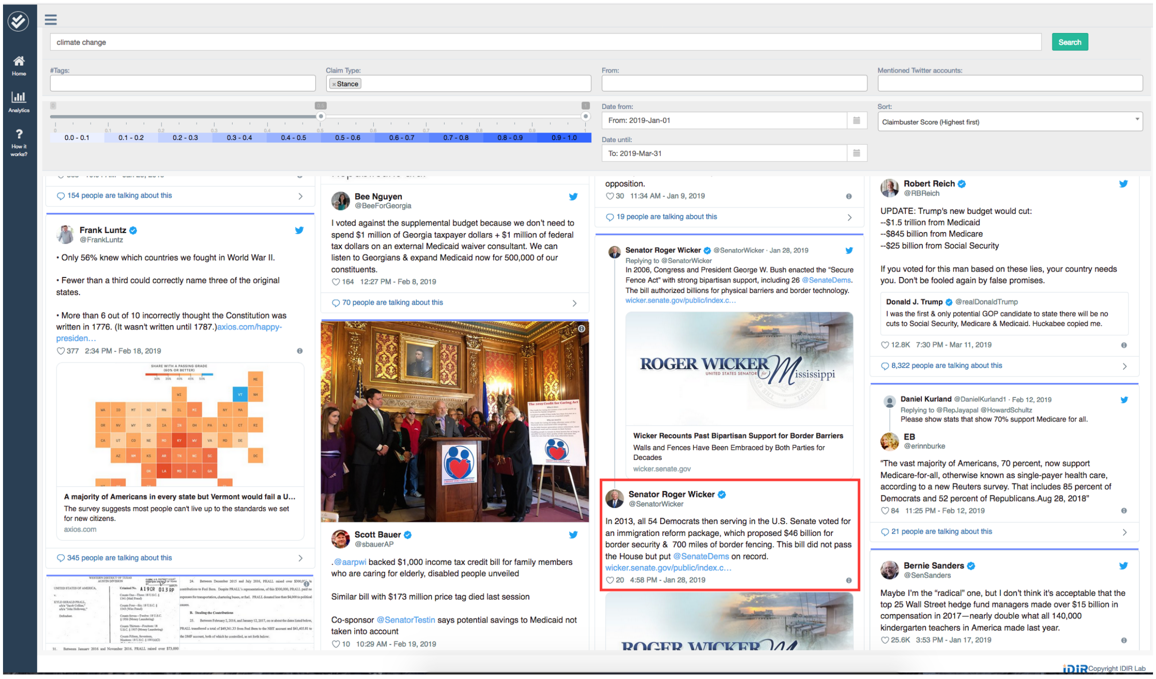

Figure 2: ClaimPortal user interface.

Claim Type	FrameNet Frames
Conflict	Invading, Attack, Explosion, Destroying, Hostile encounter, Use firearm, Shoot projectiles, Downing, Protest, Political actions
Life	Giving birth, Being born, Death, Killing, Forming relationships, Cause harm, Personal relationship, Dead or alive
Movement	Self motion, Inhibit movement, Travel, Departing, Arriving, Visiting, Motion, Cause motion, Bringing
Transaction	Import export scenario, Commerce buy, Commerce sell, Getting, Commerce pay, Borrowing, Giving
Business	Activity start, Conquering, Endeavor failure, Intentionally create, Business closure, Locale closure
Contact	Meet with, Discussion, Come together, Communication, Contacting, Communication means, Text creation, Request
Personnel	Take place of, Get a job, Hiring, Appointing, Removing, Firing, Quitting, Choosing, Becoming a member, Change of leadership
Justice	Arrest, Imprisonment, Detaining, Extradition, Breaking out captive, Try defendant, Pardon, Appeal, Verdict, Sentencing, Fining, Execution, Releasing, Notification of charges
Comparison	Comparing two entities, Comparing at two different points in time
Quantity	Change position on scale, Creating, Causation, Cause change of position on a scale, Occupy rank, Ratio
Stance	Taking sides, Opinion, Be in agreement on assessment, Vote, Oppose and Support Consistency
Speech	Statement, Affirm or deny, Telling

Table 1: Claim types and their corresponding FrameNet frames. Frames in red color are created by us.

followed it by grouping together semantically and syntactically similar factual claims from these fact-checks. If a group of claims did not share a common existing frame, we created a new frame for it. Details of these purposely created new frames can be found in (Arslan et al., 2019). The corpus of the newly-defined frames along with their annotated exemplary sentences and a paper that explains each frame in detail are publicly available. [7]

We used open-sesame (Swayamdipta et al., 2017), a recurrent neural network based frame- semantic parser, to detect all possible frames a tweet can potentially hold. We retrained open-sesame on FrameNet 1.7 dataset after extending it with annotated sentences for the newly defined frames. Open-sesame works as a pipeline of several tasks: target identification (detecting all lexical units), frame identification (detecting all frames in a sentence), and argument identification.

3.4.2 Claim type mapper

In (Spiliopoulou et al., 2017) eight ACE event types were listed along with their mapped frames: *Business, Conflict, Contact, Justice, Life, Movement, Personnel,* and *Transaction*. To accommodate the new frames explained in Section 3.4.1, we extended this list by introducing four new event types, namely *Comparison, Quantity, Stance*, and *Speech*, and their corresponding frames (Table 1). In ensuing discussion, we refer to these event types as *claim types*, for simplicity of terminology. More specifically, *Comparison* is for claims that show entities involved in some sort of comparisons based on some criteria, *Quantity* presents claims with quantities, *Stance* is for claims that have en-

[7] https://github.com/idirlab/factframe

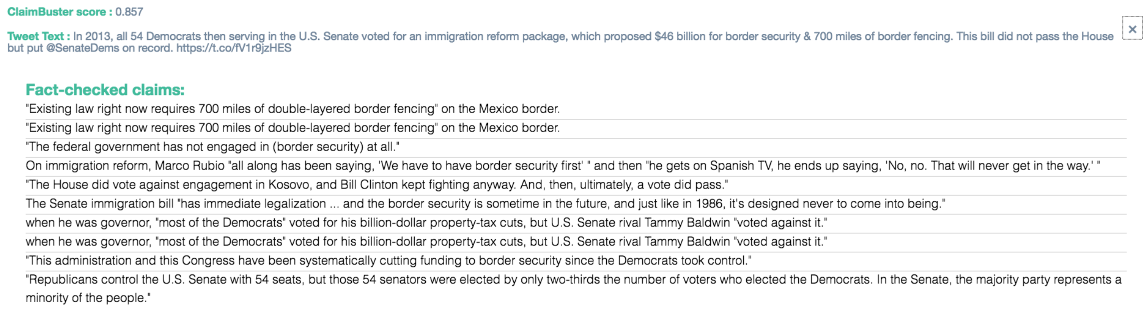

Figure 3: Similar fact-checks for the highlighted tweet in Figure 2.

tities with viewpoints towards issues, events, etc., and *Speech* is for claims that communicate some messages in the written or spoken form. A script identifies the claim types of each tweet by mapping identified frames to their corresponding claim types. A tweet can have multiple claim types.

3.5 Claim Matcher

Claim matching is an important step in the workflow of fact-checking. Given a factual claim, it aims at finding identical or similar claims from a repository of existing fact-checks. The premise is that public figures keep making the same false claims. While politicians may refrain themselves from making outright false claims to avoid being fact-checked, oftentimes they even double down after their false claims are debunked. [8]

ClaimPortal leverages the claim matching function in the ClaimBuster API. The fact-check repository was curated from various fact-checking websites. The system measures the similarity between a claim and a fact-check based on the similarity of their tokens. An Elasticsearch server is deployed for searching the repository based on token similarity.

4 User Interface Features

ClaimPortal enables the user to sift through the tweets using multiple filters. The important filters are as follows: **(1) Keyword Search:** Allows users to make a text based search by typing the desired keywords like "climate change" in the search input area at the top. This displays all the tweets pertaining to the "climate change". **(2) Tags:** Allows users to further filter tweets by hash-tags such as #116thCongress or #2020. **(3) Claim Type:** ClaimPortal enables users to search tweets based on a specific claim type (e.g., *"CONFLICT"* or *"STANCE"*). **(4) From:** The users can browse tweets created by a particular user handle (e.g., @realDonaldTrump). **(5) Mentions:** The search results can be filtered further by user mentions in a tweet (i.e., using '@' to tag a user in a tweet, such as @POTUS). **(6) ClaimBuster Score:** ClaimPortal also offers a slider interface to filter results based on a ClaimBuster score range - for example from 0.3 to 0.7. The results are automatically updated as the slider moved. **(7) Date Range:** Additionally, the portal offers a date picker to filter tweets based on their creation date.

Figure 2 shows ClaimPortal user interface with search results of a sample query. The sample query contains the following filtering conditions: a keyword 'climate change', a claim type '*Stance*', a range of ClaimBuster score between 0.5 to 1.0, and a date range from January 1, 2019 to March 31, 2019. Moreover. the ClaimPortal shares previously fact-checked claims with users by displaying matching fact-checks after clicking a tweet's card view. Figure 3 depicts the matching fact-checks of the highlighted tweet in Figure 2.

5 Analytics and Visualizations

We work to make ClaimPortal the repository where one can find all factual claims made on Twitter. It can be a powerful tool for a diverse group of users. It enables the web users to explore and analyze the factual claims in tweets at scale. We use analytics and visualizations to shed more light on the importance of ClaimPortal and bring the hidden patterns in the data to light. For instance, a user can compare tweets from different political groups in detail based on check-worthiness of their claims and variety of their claims. Figure 4 (d) and (e) compare Democratic Senators and Republican Senators based on the

[8] https://wapo.st/2rucTq8

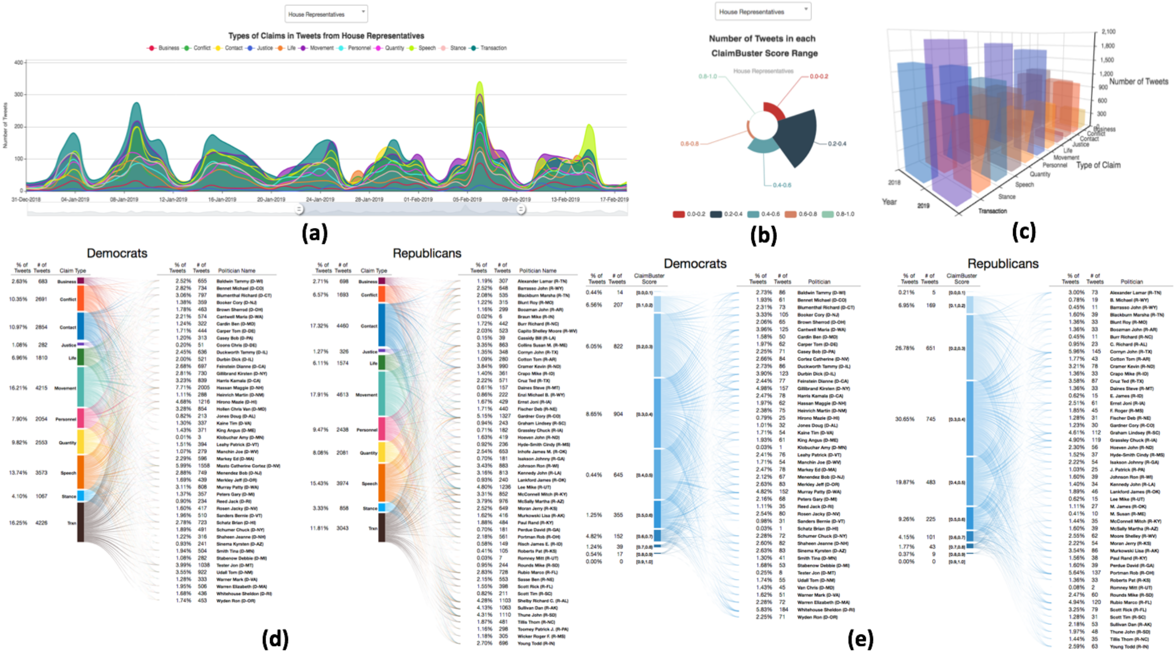

Figure 4: Examples of visualizations on ClaimPortal website.

types of claims they made and check-worthiness of their claims. Figure 4 (a) depicts the spread of all claim types made by different group of politicians in the past one year and Figure 4 (b) shows the distribution of tweets based on five ClaimBuster score range made by different group of US politicians such as 2020 Presidential candidates.

References

Bill Adair, Mark Stencel, Cathy Clabby, and Chengkai Li. 2019. The human touch in automated fact-checking: How people can help algorithms expand the production of accountability journalism. In *Proceedings of the 2019 Computation+Journalism Symposium*.

Fatma Arslan, Damian Jimenez, Josue Caraballo, Gensheng Zhang, and Chengkai Li. 2019. Modeling factual claims by frames. In *Proceedings of the 2019 Computation+Journalism Symposium*.

Mevan Babakar and Will Moy. 2016. The state of automated factchecking. *Full Fact 28*.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *ACL*, pages 86–90.

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

Sarah Cohen, Chengkai Li, Jun Yang, and Cong Yu. 2011. Computational journalism: A call to arms to database researchers. In *CIDR*, pages 148–151.

Naeemul Hassan, Bill Adair, James T. Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. 2015a. The quest to automate fact-checking. In *Proceedings of the 2015 Computation+Journalism Symposium*.

Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017a. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *KDD*, pages 1803–1812.

Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015b. Detecting check-worthy factual claims in presidential debates. In *CIKM*, pages 1835–1838.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li, and Mark Tremayne. 2017b. Claimbuster: The first-ever end-to-end fact-checking system. *PVLDB*, 10(12):1945–1948.

Damian Jimenez and Chengkai Li. 2018. An empirical study on identifying sentences with salient factual statements. In *IJCNN*.

Saehan Jo, Immanuel Trummer, Weicheng Yu, Daniel Liu, and Niyati Mehta. 2018. The factchecker: Verifying text summaries of relational data sets.

Baoxu Shi and Tim Weninger. 2016. Discriminative predicate path mining for fact checking in knowledge graphs. *Knowledge-Based Systems*, 104(C):123–133.

Evangelia Spiliopoulou, Eduard Hovy, and Teruko Mitamura. 2017. Event detection using frame-semantic parser. In *Proceedings of the Events and Stories in the News Workshop*, pages 15–20.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. *CoRR*, abs/1706.09528.

You Wu, Pankaj K. Agarwal, Chengkai Li, Jun Yang, and Cong Yu. 2017. Computational fact checking through query perturbations. *ACM Transactions on Database Systems (TODS)*, 42(1):4:1–4:41.