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

Anonymous Author(s)

ABSTRACT

Our society is struggling with an unprecedented amount of falsehood which is far more than what fact-checkers can keep up with. The problem of unchecked claims is exacerbated on social media. We present ClaimPortal, a web-based platform for providing monitoring, searching, checking, and analytics of factual claims on Twitter. We explain the architecture of ClaimPortal, its components and functions, and the user interface, as well as a plan for presenting the demonstration. 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 [5]. 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 continuously collects tweets and *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, trending claims, and so on.

We leverage NLP techniques, high-performance database storage and querying techniques, and deep learning models to assist us in our goal of delivering an innovative tool. To elaborate, the check-worthiness scores are computed by an SVM model [6] although we expect to soon deploy a newer deep learning model based on adversarial training methods [11]. Our back-end is supported by a MySQL database as well as the Elasticsearch search engine. These combined provide efficient storage and retrieval of our data for our front-end to display. Finally, for matching claims with their types we use a segmental recurrent neural network (SRNN) [14] trained on a manually labeled data-set. Together these state-of-the-art tools and techniques allow us to build a robust and innovative system.

We aim to make ClaimPortal the repository where one can find all factual claims made on Twitter. It can be a highly powerful tool to various types of users. It empowers *general web users* to explore and analyze the factual claims in tweets at scale and thus catalyzes democratized data analytics and fact-checking. It assists *professional fact-checkers and journalists* in covering an important but often overlooked arena of fact-checking. It will become an important infrastructure and data resource to the *interdisciplinary community of*

WWW 2019, May 13-17, 2019, San Francisco, United States

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

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

researchers in areas related to computational fact-checking, including journalism, communication studies, psychology, and political science, in addition to computer science.

While the initial call to arms to research on computational fact-checking was made nearly a decade ago [4], 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 [7, 9], to using databases for discerning factual claims' robustness [15] and truthfulness [3, 10, 12], to building end-to-end fact-checking systems [1, 6, 8]. ClaimPortal is a novel infrastructure in that fact-checkers have largely skipped factual claims in tweets.

The success of fact-checking lies not only in methodology and technology but also education. ClaimPortal will also be an educational resource in helping cultivate a society that is less susceptible to falsehoods, raise the awareness of all aspects of misinformation and disinformation, and train a generation of Web users that are well versed in media literacy, data literacy, logic, and fallacy.

2 SYSTEM ARCHITECTURE, COMPONENTS, AND USER INTERFACE

2.1 System Architecture

ClaimPortal is composed of an MySQL database, an Elasticsearch[2] search engine, a front-end GUI and several decoupled data processing programs (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 is a faceted search interface allowing users to narrow down search results by applying multiple filters based on the faceted classification of items. Keyword search on tweets is powered by Elasticsearch which is coupled with querying the database to provide additional filters. The back-end layer consists of the data gathering process, which does tweet pre-processing and computing check-worthiness scores of tweets using the public ClaimBuster API [6], as well as Elasticsearch batch insertion. ClaimPortal stays up-to-date with current tweets by periodically calling the Twitter REST API, particularly, the user_timeline resource.

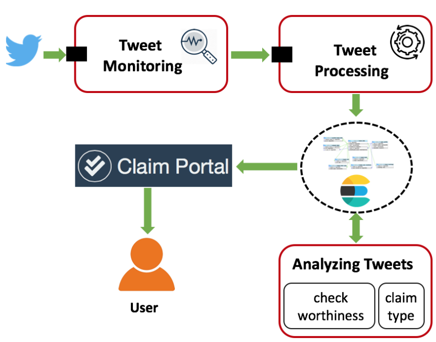

Figure 1: ClaimPortal **system architecture**

2.2 Tweet Monitoring, Processing, and Storing

ClaimPortal focuses on politically-charged tweets. We curated a list of prominent Tweet handles in U.S. politics that include but are not limited to 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. Focusing on essential politicians and users on Twitter not only gives us much cleaner data but also provides important and relevant tweets. First, we navigated through the historic data of a user's timeline, which is a one-time process, and then we keep our data up-to-date by continuously monitoring newly posted tweets. Currently, ClaimPortal monitors 3,200 Twitter handles and has collected around 2.6 million tweets after being deployed for a couple of weeks. We are also working on 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, we do 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.

Before any collection begins, we first configure the MySQL database with several normalized tables. We store tweet-text, when it was created, and who tweeted it. We also store information regarding re-tweets and quoted-tweets, hash-tags and URLs mentioned in each tweet, and information related to the accounts mentioned in the tweet.

We use Elasticsearch for supporting keyword search. Since Elasticsearch is equipped with incremental indexing, each time we supply the delta tweets since the last update to Elasticsearch. For this we have a decoupled process that takes care of Elasticsearch bulk insertion which significantly speeds up the updating of the index.

2.3 Check-worthiness Score

Each tweet's text is associated with a ClaimBuster score which reflects how check-worthy it is in terms of being a factual claim. This score is obtained by utilizing the ClaimBuster API [6]. ClaimBuster is a SVM-based classification model trained on 5,000 sentences that were labeled as check-worhty, not check-worthy, or non-factual. It finds factual claims from natural language sentences. The ClaimBuster API returns a check-worthiness score for a given text. The score is on a scale of 0 to 1 where 0 means a claim is less likely to be check-worthy and 1 means it is very likely to be check-worthy. This helps us to rank the tweets after a search query in the GUI. Obtaining check-worthiness score of tweets is done in parallel to the tweet collection and Elasticsearch bulk insertion processes.

2.4 Detecting Claim Types

ClaimPortal uses tweets to gain insights into what ideas are being spread, by who, how often, and their veracity. To facilitate this we classify tweets by the types of factual claims in the tweets. This enables statistics on what types of claims are being made, what they concern, and when they gained momentum. We employed a collection of FrameNet [2] frames for detecting tweets' claim types.

FrameNet is a linguistic resource for English comprising of 1,224 manually established semantic frames. Each frame provides information about both the linguistic and semantic structure of a type of

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

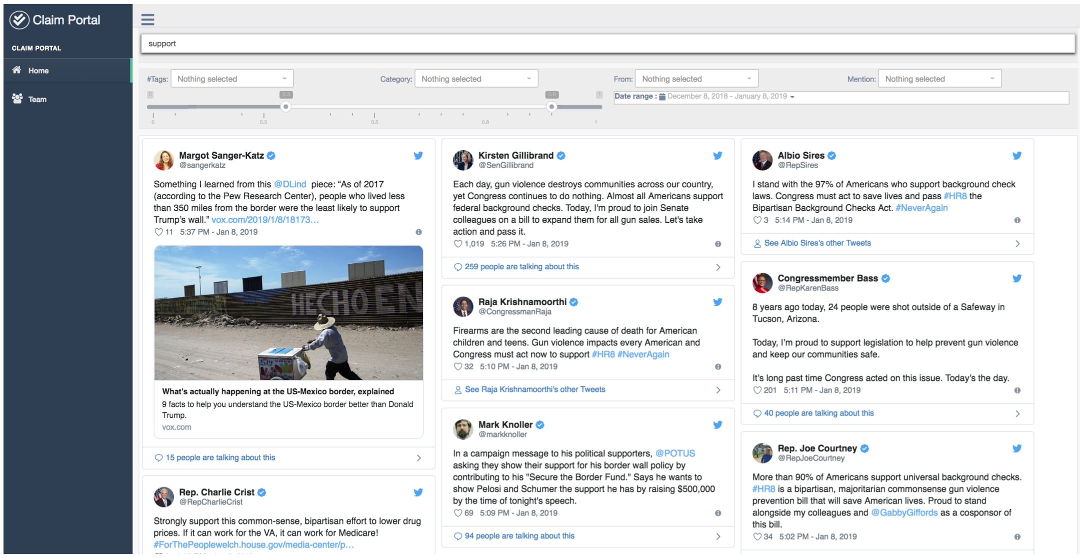

Figure 2: ClaimPortal **user interface**

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

Claim Type	FrameNet Frames
PERSONNEL	Take place of, Get a job, Hiring, Appointing, Removing, Firing, Quitting, Choosing, Becoming a member, Change of leadership
TRANSACTION	Import export scenario, Commerce buy, Commerce sell, Getting, Commerce pay, Borrowing, Giving
COMPARISON	Comparing two entities, Comparing at two different points in time
SPEECH	Statement, Affirm or deny, Telling
STANCE	Taking sides, Opinion, Be in agreement on assessment, Vote, Oppose and Support Consistency
QUANTITY	Change position on scale, Creating, Causation, Cause change of position on a scale, Occupy rank, Ratio

event, situation, object, or relation along with its participants. The participants, called Frame Elements, are frame-specific semantic roles providing additional information. Each frame is evoked by a set of lexical units (LUs), or words, which are a composition of the lemma and meaning of the word.

Each claim type contains several frames which convey the same broad meaning. We adopted the event categories introduced in [13] as claim types. Their study presents eight ACE event categories and mapped frames for each category. These categories are: BUSINESS, CONFLICT, CONTACT, JUSTICE, LIFE, MOVEMENT, PERSONNEL, and TRANSACTION. We extended these claim types by introducing four new categories and their corresponding frames. (Table 1)

We collected fact-checked claims from PolitiFact[3] and chose a number of these sentences to thoroughly examine one-by-one. We then grouped claims by finding the semantic and syntactic similarities between them. At the end of this process we had four groups of claims. For the claim groups where we found suitable FrameNet frames, we simply used them. For the remaining claims in these groups where we felt either FrameNet did not have a suitable frame, or its frames did not capture what we thought they should, then a new frame or frames were created. Therefore, we have designed five new frames that are novel in what they capture and are created with factual claims specifically in mind. Overall, we have 12 claim types and 86 corresponding frames. Table 1 presents PERSONNEL and TRANSACTION adopted categories and their corresponding frames and four new categories which are COMPARISON, SPEECH, STANCE, and QUANTITY.

We used open-sesame [14], a recurrent neural network based frame-semantic parser, to detect all possible frames a tweet holds. Open-sesame works as a pipeline of several tasks including, target identification task, frame identification task, and argument identification task. Currently, we are only using the target identification task which detects all LUs and the frame identification task which detects all frames of a tweet. We then identify the claim types of each tweet based on identified frames.

2.5 Claim Matching

Claim matching is an important step in the workflow of fact-checking. Given a factual claim, this step aims at matching a claim against a repository of existing fact-checks. The premise is that politicians and 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. [4]

In the simplest case, when someone repeats a claim identical to one that has been fact-checked, the existing fact-check's verdict can be presented to fact-checkers for further investigation and directly

[3]https://www.politifact.com

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

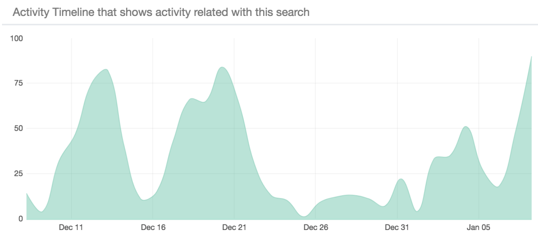

Figure 3: Number of tweets on "support"

to news audience as well. An example of the former case is Full Fact's in-house platform. [5] These approaches can all leverage the tens of thousands of fact-checks produced by various organizations over the years. To leverage some of this data we have employed the "Share the Facts" dataset in our system to find matching fact-checks.

2.6 User Interface Features

ClaimPortal empowers the user with a faceted search interface which enables sifting through tweets using multiple filters. The default view shows the most current tweets that are being monitored by ClaimPortal, with at a ClaimBuster score of at least 0.3. We will now explain how easy it is to use our interface using a combination of the following filters:

Keyword Search Users can make a text based search by typing the desired keywords, for instance we can enter "vote" in the search input area at the top. This will give us all the tweets that are talking about "vote".

ClaimBuster Score With the power of faceted interface, users can make use of the slider interface to filter results based on a ClaimBuster score range. Lets say, we make a score selection range from 0.3 to 0.9. The results will be automatically updated as we move the slider.

Date Range Additionally, a date picker interface is provided to filter using the creation date of a tweet. If we select a date range from December 8, 2018 to January 8, 2019, we will only see tweets created in this date range.

Mentions We can filter the results further by user mentions in a tweet (i.e., using @ to tag or "at" another user, such as @POTUS or @BarackObama).

Hash-Tags The portal also provides users with filtering tweets that contain specific hash-tags such as #116thCongress or #2020.

User Handle The interface allows browsing tweets that are created by a particular user handle (e.g., @realDonaldTrump).

Claim Type Finally, an option is provided to filter the search results based on a specific claim type (e.g., "CONFLICT" or "STANCE").

Figure 2 shows the faceted interface with tweet results to a sample search query. In this sample query, we used keyword 'support', a range of check-worthiness score between 0.3 to 0.9, and a date range from December 8, 2018 to January 8, 2019. Users are able to visualize their search results in the form of a line chart and pie chart to display the number of tweets over time and popular hash-tags as shown in Figure 3 and 4, respectively, for the sample search query.

Matching claims We can also see matching fact-checks to tweet by clicking its card view.

5 https://fullfact.org/

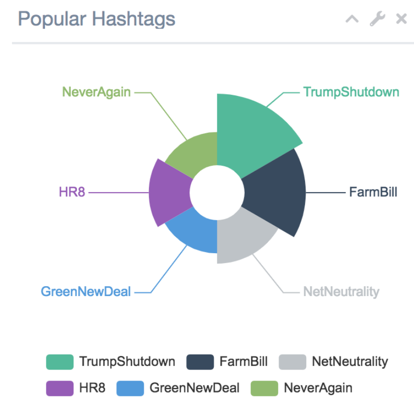

Figure 4: Popular hash-tags after search query on "support"

REFERENCES

[1] Mevan Babakar and Will Moy. 2016. The state of automated factchecking. *Full Fact 28* (2016).

[2] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *ACL*. 86–90.

[3] 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 (2015), 1–13. Issue 6.

[4] Sarah Cohen, Chengkai Li, Jun Yang, and Cong Yu. 2011. Computational Journalism: A Call to Arms to Database Researchers. In *CIDR*. 148–151.

[5] Naeemul Hassan, Bill Adair, James T. Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. 2015. The Quest to Automate Fact-Checking. In *Proceedings of the 2015 Computation+Journalism Symposium*.

[6] Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. Toward Automated Fact-Checking: Detecting Check-worthy Factual Claims by ClaimBuster. In *KDD*. 1803–1812.

[7] Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015. Detecting Check-worthy Factual Claims in Presidential Debates. In *CIKM*. 1835–1838.

[8] 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. 2017. ClaimBuster: The First-ever End-to-end Fact-checking System. *PVLDB* 10, 12 (Aug. 2017), 1945–1948.

[9] Damian Jimenez and Chengkai Li. 2018. An Empirical Study on Identifying Sentences with Salient Factual Statements. In *IJCNN*.

[10] Saehan Jo, Immanuel Trummer, Weicheng Yu, Daniel Liu, and Niyati Mehta. 2018. The FactChecker: Verifying Text Summaries of Relational Data Sets. *arXiv preprint arXiv:1804.07686*.

[11] Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2017. Adversarial Training Methods for Semi-Supervised Text Classification. *ICLR* (2017).

[12] Baoxu Shi and Tim Weninger. 2016. Discriminative Predicate Path Mining for Fact Checking in Knowledge Graphs. *Knowledge-Based Systems* 104, C (2016), 123–133.

[13] 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*. 15–20.

[14] 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 (2017).

[15] 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 (2017), 4:1–4:41.