ClaimBuster:
The First-ever End-to-end Fact-checking System

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 [♯] Mark Tremayne [♮]

[§] Department of Computer and Information Science, University of Mississippi
[♯] Department of Computer Science and Engineering, University of Texas at Arlington
[♮] Department of Communication, University of Texas at Arlington

nhassan@olemiss.edu, {gensheng.zhang, fatma.dogan, josue.caraballo, damian.jimenez, siddhant.gawsane}@mavs.uta.edu
{shohedul.hasan, minumol.joseph, aaditya.kulkarni, anil.nayek, vikas.sable}@mavs.uta.edu, {cli, tremayne}@uta.edu

1. INTRODUCTION

Our society is struggling with an unprecedented amount of falsehoods, hyperboles, and half-truths. Politicians and organizations repeatedly make the same false claims. Fake news floods the cyberspace and even allegedly influenced the 2016 election. In fighting false information, the number of active fact-checking organizations has grown from 44 in 2014 to 114 in early 2017. [1] Fact-checkers vet claims by investigating relevant data and documents and publish their verdicts. For instance, PolitiFact.com, one of the earliest and most popular fact-checking projects, gives factual claims truthfulness ratings such as True, Mostly True, Half true, Mostly False, False, and even "Pants on Fire". In the U.S., the election year made fact-checking a part of household terminology. For example, during the first presidential debate on September 26, 2016, NPR.org's live fact-checking website drew 7.4 million page views and delivered its biggest traffic day ever.

The challenge is that the human fact-checkers cannot keep up with the amount of misinformation and the speed at which it spreads. One of the reasons for this is that fact-checking is an intellectually demanding, laborious, and time-consuming process. This challenge creates an opportunity for automated fact-checking systems. On the other hand, fact-checking technology is clearly falling behind, as there is simply no existing system that truly does automated fact-checking. Today's professional fact-checkers diligently perform their work as an art, following good practices in data and investigative journalism. A recent white paper [2] surveys existing tools that can be integrated. Although the relevant tools and techniques can assist fact-checking in various areas, a full-fledged, end-to-end solution does not exist. There have been some attempts,[2] but those efforts did not lead to such fact-checking systems.

*Work performed while at UT-Arlington.

[1] http://reporterslab.org/fact-checking/

[2] T. Wilner. *Fail and move on: Lessons from automated fact-checking experiments.* Poynter, September 7, 2016. http://goo.gl/G0l54Y

Proceedings of the VLDB Endowment, Vol. 10, No. 12

Starting in December 2014, we have been building ClaimBuster, an end-to-end system that uses machine learning, natural language processing, and database query techniques to aid in the process of fact-checking. It monitors live discourses (e.g., interviews, speeches and debates), social media, and news to identify factual claims, detect matches with a curated repository of fact-checks from professionals, and deliver those matches instantly to the audience (e.g., by displaying a pop-up warning if a presidential candidate makes a false claim during a live debate). For various types of new claims not checked before, ClaimBuster automatically translates them into queries against knowledge databases and reports whether they check out. For claims where humans must be brought into the loop, it provides algorithmic and computational tools to assist lay persons and professionals in understanding and vetting the claims. Its use will be expanded to verify both political and non-political claims in many types of narratives, discourses, and documents such as sports news, legal documents, and financial reports.

While the development of the full-fledged system is still ongoing, several components of ClaimBuster are integrated and deployed in the real-world. One of its most mature components, the *claim spotter*, discovers factual claims that are worth checking. Given the plethora of discourses and narratives we are constantly exposed to, ClaimBuster gives each sentence a score that indicates how likely it contains an important factual claim that should be checked. This essentially provides a priority ranking that helps fact-checkers efficiently focus on the top-ranked sentences without painstakingly sifting through a large number of sentences.

ClaimBuster was tested in real-time during the live coverage of all primary and general election debates throughout the 2016 U.S. election. Closed captions of the debates on live TV broadcasts, captured by a decoding device, were fed to ClaimBuster, which immediately scored each sentence spoken by the candidates and posted top-scored claims to the project's website (http://idir.uta.edu/ClaimBuster) and Twitter account (@ClaimBusterTM). Post-hoc analysis of the claims checked by professional fact-checkers at CNN, PolitiFact.com, and FactCheck.org reveals a highly positive correlation between ClaimBuster and journalism organizations in deciding which claims to check [5]. ClaimBuster has also been continuously monitoring Twitter and retweeting the check-worthy factual claims it finds in tweets from politicians and organizations (twitter.com/ClaimBusterTM). Recently it started to monitor "Hansard" [3] – the transcripts of proceedings of the Australian parliament (http://idir.uta.edu/ClaimBuster/hansard).

[3] http://www.aph.gov.au/Parliamentary_Business/Hansard

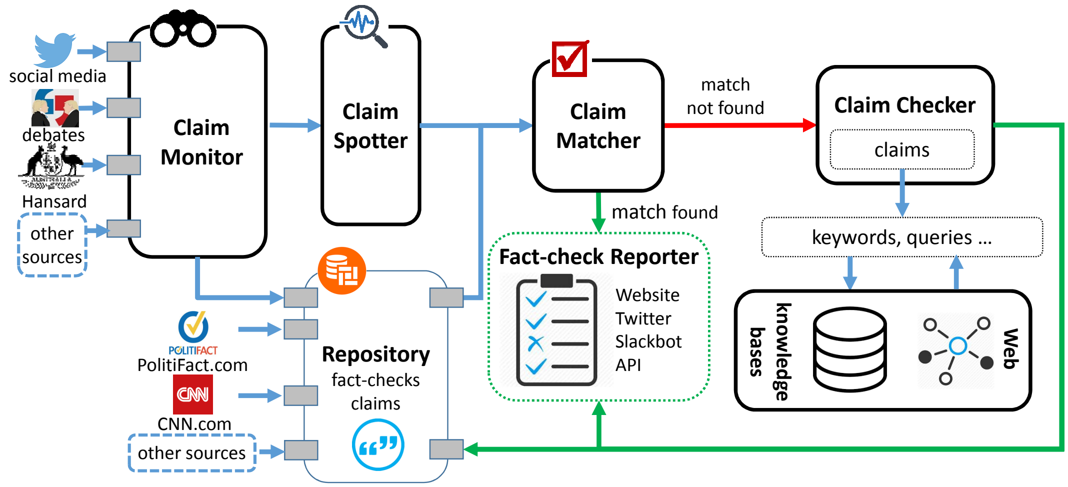

Figure 1: System architecture of ClaimBuster.

ClaimBuster already produces true-or-false verdicts for certain types of factual claims. Given a factual claim which is scored highly by the claim spotter component, ClaimBuster may reach a verdict by two methods. One method is to translate the factual claim into questions and their accompanying answers. It then sends the questions to question-answering systems and compares the returned results with the aforementioned answers. It produces a verdict based on the presence/absence of a discrepancy between these two sets of answers. The other method is to search in a repository for similar or identical claims that have already been fact-checked by professionals and to use the verdicts from the professionals. In the case that ClaimBuster is not able to produce a verdict, it provides processed search results from a general search engine to assist vetting the claim.

The ClaimBuster project has received wide recognition in the fact-checking community and substantial media coverage. (See http://idir.uta.edu/claimbuster/press for a list of media outlets and the stories in which they cover ClaimBuster.) The aforementioned white paper calls ClaimBuster a tool with "the most advanced generalised automatic claim spotting." [2] Others considered it "perhaps the biggest development to date" in ranking claims [4] and "a pretty useful guide for journalists and those members of the public who wish to spend time using an algorithm to help find facts." [5]

ClaimBuster, upon completion, can benefit a large base of potential users. It directly benefits citizens and consumers by improving information accuracy and transparency. It helps news organizations speed up their fact-checking and ensure the accuracy of their news stories. Businesses can use ClaimBuster to identify falsehoods in their competitors' and their own reports and press releases. It can also assist professionals such as lawyers in verifying documents.

2. SYSTEM OVERVIEW

The ClaimBuster system is hosted at http://idir.uta.edu/ClaimBuster and its features are being constantly expanded. Figure 1 depicts its system architecture. The *claim monitor* interfaces various data sources (social media, broadcasted TV programs, and websites) with ClaimBuster. The *claim spotter* identifies check-worthy factual claims in verbose text from the data sources. The *claim matcher* finds existing fact-checks that are closely-related or identical to the discovered claims. In this way, we fully leverage well-researched fact-checks from professional fact-checkers. This is particularly useful, because oftentimes the same false claims are repeated. [6] When a matching fact-check cannot be found, the *claim checker* queries external knowledge bases and the Web to vet the factual claims. The *fact-check reporter* compiles the evidence from the claim matcher and the claim checker, and presents fact-check reports to users through various channels, such as the project website, its Twitter account, a Slackbot, and a public API. Below we explain these components in more detail.

Claim Monitor: This component continuously monitors and retrieves texts from a variety of sources, upon which claim spotting is applied to discover important factual claims. At present, the system monitors the following sources.

Broadcast Media: ClaimBuster uses a decoding device to extract closed captions in broadcasted TV programs. This was used for our live coverage of all twenty-one primary election debates and four general election debates of the 2016 U.S. presidential election. One challenge in delivering the live coverage of such events is the lack of speaker identity in the closed captions. ClaimBuster timely derives the speaker of a sentence using the Twitter Streaming API. [7] The idea is based on the premise that, during a popular live event, active Twitter users tend to mention the speaker while tweeting a statement the speaker made. Details of speaker identification in ClaimBuster can be found in [7].

Social Media: ClaimBuster has been continuously monitoring a list of 2220 Twitter accounts (U.S. politicians, news and media organizations) using the Twitter streaming API. It filters out non politics-related tweets using an SVM classifier [1].

Websites: ClaimBuster also gathers data from websites. For instance, as mentioned in Section 1, it monitors the transcripts of proceedings of the Australian parliament.

Claim Spotter: Given a sentence, ClaimBuster gives it a score between 0.0 and 1.0. The higher the score, the more likely the sentence contains check-worthy factual claims. The lower the score, the more non-factual, subjective and opinionated the sentence is. ClaimBuster's score is based on a classification and scoring model. The model was trained using tens of thousands of sentences from past general election debates that were labeled by human coders.

[4] K. Moreland and B. Doerrfeld. *Automated Fact Checking: The Holy Grail of Political Communication.* Nordic APIs, February 25, 2016. http://goo.gl/uhsnyT

[5] P. Fray. *Is that a fact? Checking politicians' statements just got a whole lot easier.* The Guardian, April 18, 2016. http://goo.gl/1UJfzU

[6] A. D. Holan. *All Politicians Lie. Some Lie More Than Others.* The New York Times, December 11, 2015. http://goo.gl/Js0XGg

[7] http://dev.twitter.com/streaming/overview

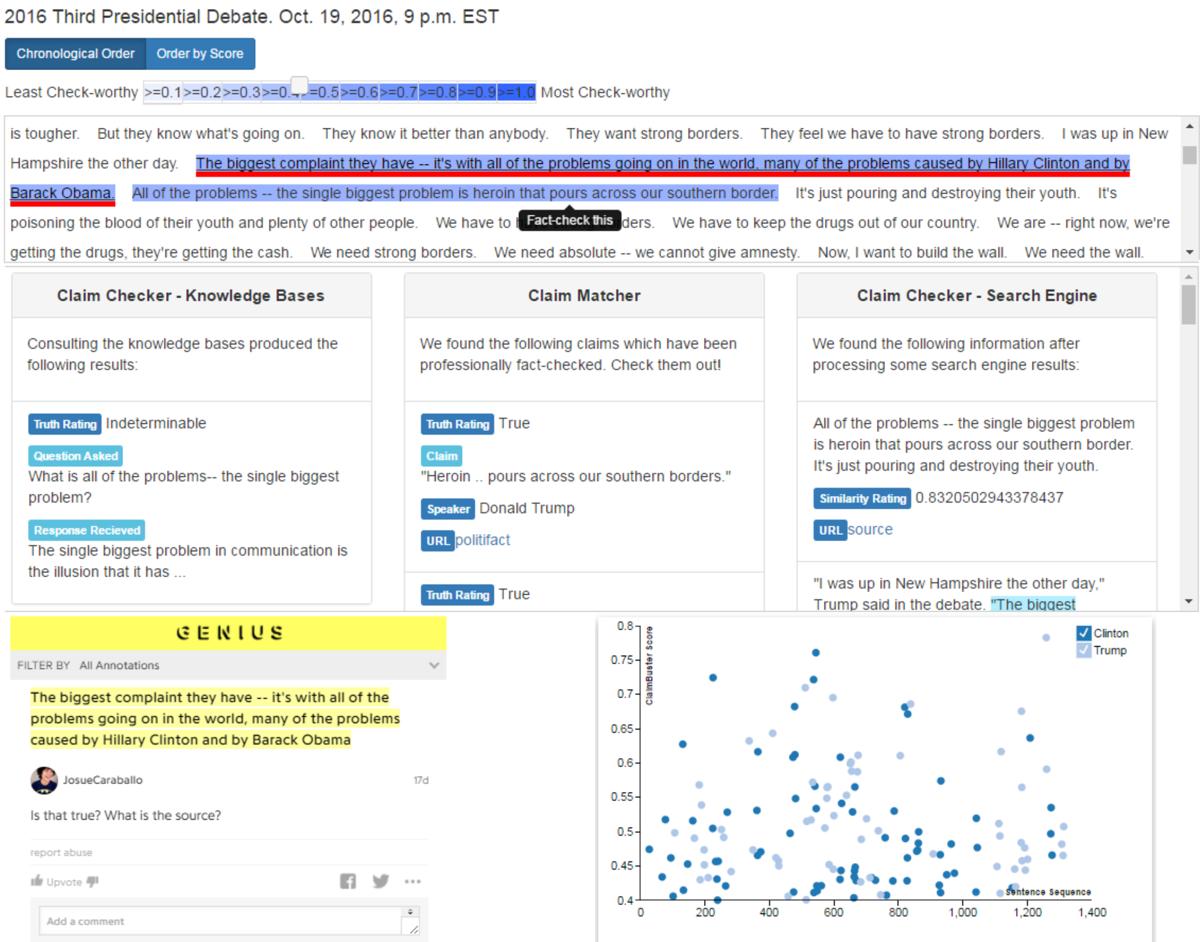

Figure 2: The user interface of ClaimBuster when it is applied on a debate.

Its features include the tokens in sentences and the tokens' part-of-speech (POS) tags. The recall and precision in detecting check-worthy factual claim are 74% and 79%, respectively [3, 4].

The claim spotter has been applied on sentences from all the aforementioned sources, including the closed captions of the presidential debates, tweets, and Hansard. Post-hoc analysis of the claims from the primary debates for the 2016 U.S. presidential election checked by professional fact-checkers at CNN, PolitiFact.com and FactCheck.org reveals a highly positive correlation between ClaimBuster and journalism organizations in deciding which claims to check and the topics of the selected claims [5]. Although its scoring and ranking model was trained using a labeled dataset of presidential debates, we find that the model achieved strong results on politics-related tweets and Hansard as well.

Claim Matcher: Given an important factual claim identified by the claim spotter, the *claim matcher* searches a fact-check repository and returns those fact-checks matching the claim. The repository was curated from various fact-checking websites. The system has two approaches to measuring the similarity between a claim and a fact-check. One is based on the similarity of tokens and the other is based on semantic similarity. An Elasticsearch [8] server is deployed for searching the repository based on token similarity, while a semantic similarity search toolkit, Semilar [8], is applied for the search based on semantic similarity. We combine the search results from both in finding fact-checks similar to the given claims.

Claim Checker: Given a claim, the *claim checker* collects supporting or debunking evidence from knowledge bases and the Web. With regard to knowledge bases, it uses a question generation tool [6] to generate many questions based on the claim and select those *good* questions which are then sent to the question answering engine Wolfram Alpha via an API. [9] Then the answers from Wolfram Alpha are extracted. Simultaneously, it sends the aforementioned questions to Google via `HTTP` requests and extracts the answers from Google's answer boxes in the `HTML` responses. If any clear discrepancies between the returned answers and the claim exist, then a verdict may be derived and presented to the user.

Meanwhile, the factual claim itself is sent to Google as a general search query. The claim checker then parses the search result and downloads the web page for each top result. Within each such page, it finds sentences matching the claim. The matching sentences and a few of their surrounding sentences are then grouped together into a *context*. The contexts, answers returned from Wolfram Alpha and Google answer boxes, as well as any verdicts derived from those answers form the supporting or debunking evidence for the claim. The evidence is reported to the user, as follows.

Fact-check Reporter: The *fact-check reporter* synthesizes a report by combining the aforementioned evidence and delivers it to users through the project website. Furthermore, ClaimBuster also delivers the claim spotter scores on claims through a variety of

[8] https://github.com/elastic/elasticsearch

[9] http://products.wolframalpha.com/api/

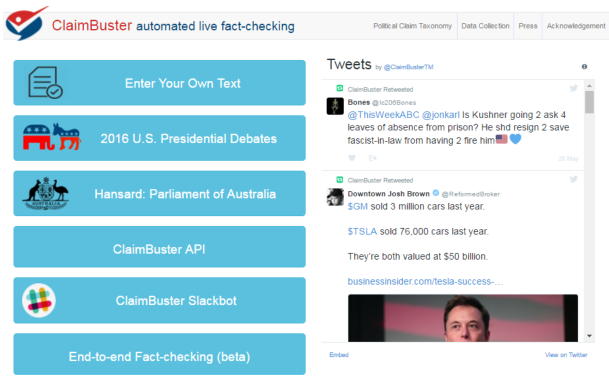

Figure 3: The homepage of ClaimBuster website.

channels, including its website, Twitter account, API, and Slackbot. Its Twitter account (@ClaimBusterTM) retweets the highly-scored tweets from politicians and organizations and posts highly-scored claims from live events such as the presidential debates. To this date, @ClaimBusterTM has retweeted and posted about $13K$ check-worthy factual claims. A Slackbot has been developed for users to supply their own text (i.e., directly as input or through a shared Dropbox folder) and receive the claim spotter score and fact-check report for that piece of text. The Slackbot has been published in the public Slack App directory and can also be installed by clicking the "ClaimBuster Slackbot" button in Figure 3. We also made available a public ClaimBuster API (note the button in Figure 3) to allow developers create their own fact-checking applications.

3. USER INTERFACE AND DEMONSTRATION PLAN

We will demonstrate the user interface features of ClaimBuster's website, Twitter account and Slackbot. Figure 3 is a screenshot of ClaimBuster's homepage. It allows a user to apply ClaimBuster on their own text or view its results on the 2016 U.S. presidential debates and the Australian Hansard (cf. Section 2). The homepage also embeds tweets from the ClaimBuster Twitter account.

Figure 2 is a screenshot of ClaimBuster applied on an archived presidential debate. The interface for the Australian Hansard is similar. Besides the basic information of the debate (e.g., title, date), the interface shows five panels. (1) The transcript panel displays the transcript of the debate. (2) The fact-check report panel displays supporting or debunking evidence collected by claim matcher and claim checker. (3) The social discussion panel allows users to discuss factual claims while collaboratively vetting them. (4) The video panel (omitted due to space limitations) has an embedded player which plays the debate video from YouTube. (5) The visualization panel shows a word cloud (omitted) and a claim spotter score chart for the sentences in the transcript.

Sentences in the transcript panel are highlighted using different shades of blue proportional to their claim spotter scores. The website allows a user to sort the sentences by time or score and to use a slider to specify the minimum score for sentences to be highlighted. Every sentence can be annotated. An annotated sentence is underlined in red. Users can discuss it, powered by the Genius platform (https://genius.com/), while collaboratively vetting it.

When a user selects a sentence in the transcript panel, the fact-check report panel displays the supporting or debunking evidence for the selected sentence. Specifically, it shows three types of evidence. The leftmost column displays similar fact-checks (along with the verdicts) from the fact-check repository, if any. The middle column shows answers extracted from Wolfram Alpha and Google answer boxes. The rightmost column displays the related search results from Google. The fact-check report is also directly available for any input sentence when the user clicks the button "End-to-end Fact-checking" in Figure 3.

The website further visualizes the content of a transcript using a word cloud and a claim spotter score chart. In the score chart, the data points represent the sentences and are color-coded by speakers. The x-axis of a point represents the corresponding sentence's sequential position in the transcript, and the y-axis corresponds to the claim spotter score of the sentence.

When a user adds the ClaimBuster Slackbot to their slack group, the user can ask the bot to fact-check a statement by using two commands. The `/getscore` command returns the claim spotter score of a sentence, e.g., Figure 4. The `/factcheck` command on a sentence returns its fact-check report.

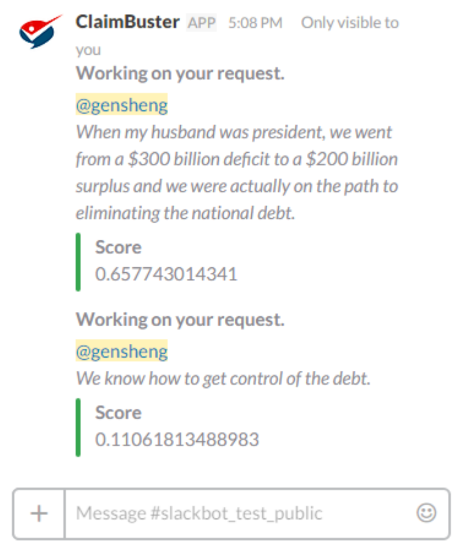

Figure 4: The ClaimBuster Slackbot.

Acknowledgements: This work is partially supported by NSF grants IIS-1408928, IIS-1565699 and a Knight Prototype Fund from the Knight Foundation. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agencies.

4. REFERENCES

[1] F. Arslan. Detecting real-time check-worthy factual claims in tweets related to U.S. politics. Master's thesis, University of Texas at Arlington, 2015.

[2] FullFact.org. The State of Automated Factchecking. *Full Fact*, August, 2016. https://fullfact.org/blog/2016/aug/automated-factchecking/.

[3] N. Hassan, B. Adair, J. T. Hamilton, C. Li, M. Tremayne, J. Yang, and C. Yu. The quest to automate fact-checking. In *Computation+Journalism Symposium*, 2015.

[4] N. Hassan, C. Li, and M. Tremayne. Detecting check-worthy factual claims in presidential debates. In *CIKM*, pages 1835–1838, 2015.

[5] N. Hassan, M. Tremayne, F. Arslan, and C. Li. Comparing automated factual claim detection against judgments of journalism organizations. In *Computation+Journalism Symposium*, 2016.

[6] M. Heilman and N. A. Smith. Question generation via overgenerating transformations and ranking. Technical report, CMU-LTI-09-013, Carnegie Mellon University, 2009.

[7] M. Joseph. Speaker identification in live events using Twitter. Master's thesis, University of Texas at Arlington, 2015.

[8] V. Rus, M. C. Lintean, R. Banjade, N. B. Niraula, and D. Stefanescu. Semilar: The semantic similarity toolkit. In *ACL*, 2013.