MODELING FACTUAL CLAIMS WITH SEMANTIC FRAMES: DEFINITIONS, DATASETS, TOOLS, AND FACT-CHECKING APPLICATIONS

by

FATMA ARSLAN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2021

To my family.

"People talk about the story of immigration as one big happy tale, but in every immigrant story there is a sadness as well, the sadness of a country, a culture, and a family that left behind, a mother who would quietly weep at night distance from the child she loved."

- Fareed Zakaria

ACKNOWLEDGEMENTS

A Ph.D. is a long and arduous journey through foreign lands, where you often encounter obstacles and search your way through deep, dark pits. I am grateful to the many people I got to meet and who accompanied me on this journey.

First and foremost, my greatest thanks go to my advisor, Dr. Chengkai Li, for his unwavering support and encouragement despite his ever-increasing responsibilities. I am eternally grateful to him for his contribution to who I am today as a researcher. His dedication to research, creativity, and perfectionism have enriched my personality as a whole. None of my achievements in graduate school would be possible without Dr. Li's guidance, inspiration, and support. He will always be a role model for me.

I have also been fortunate to have Dr. Leonidas Fegaras, Dr. Vassilis Athitsos, Prof. David Levine as my committee members, whom I thank for serving in my committee, for guiding me, and sharing their wisdom with me.

I extend my sincere appreciation to my colleagues at UTA. In particular, I am grateful to my research collaborators Dr. Naeemul Hassan, Dr. Gengsheng Zhang, Dr. Mark Tremayne, Damian Jimenez, and Josue Caraballo. I would not have been able to finish the work in this dissertation without their help and support. I also want to thank my amazing friends and labmates at the IDIR Lab, Dora, Farahnaz, Damian, Josue, Israa, Xiao, Sami, Allen, Jacob, Nasim, Penny, Joey for making the IDIR lab a home to me, for always supporting me, and for making my Ph.D. journey a rich experience.

I would like to extend my gratitude to the CSE department at UTA and Dr. Li for providing me with financial supports throughout my entire Ph.D. I would also like to thank Dr. Ramez Elmasri and the CSE department for allowing me to design and teach a course

of my own. It was a rewarding experience, and I learned a lot from student feedback. A special thanks go to Dr. Bahram Khalili for being such a great graduate advisor and for his guidance throughout my graduate studies. I want to extend my appreciation to all the excellent faculty and staff at the CSE department for their support.

I am grateful to all the teachers who taught me during the years I spent in school, first in Turkey, then in the Unites States. I would like to thank Dr. Burhan Ergen and Dr. Galip Aydin for encouraging and inspiring me to pursue graduate studies.

A special thanks to Dr. Sona Hasani, Dr. Gunes Alkan, Dr. Frahnaz Akrami, Theodora Toutountzi (soon to be a doctor), and Dr. Bhanu Jain for being my support system. Your feedback, moral support, and friendship kept me going.

I am also extremely grateful to my parents: Nahide and Eyup Dogan. Their sacrifice, encouragement, patience and prayers kept me going. Thank you mom and dad. I also want to thank my siblings for their support and love. To my friends outside of academia, thanks for keeping me sane. I am particularly obliged to Mustafa and Gizem for their everlasting friendship and support. You brought plenty of fun into my challenging Ph.D. life.

Finally, my deepest appreciation goes to my husband, my best friend, and my biggest supporter Hakan. Thank you for always being there for me, lifting me up every time I fall, encouraging me to follow my dreams, and supporting me through all the good and bad times. I would not have made it this far without your endless love and support.

July 29, 2021

ABSTRACT

MODELING FACTUAL CLAIMS WITH SEMANTIC FRAMES: DEFINITIONS, DATASETS, TOOLS, AND FACT-CHECKING APPLICATIONS

Fatma Arslan, Ph.D.

The University of Texas at Arlington, 2021

Supervising Professor: Chengkai Li

As social media sites have become major channels for the quick dissemination of news, misinformation has become a significant challenge for our society to tackle. Today fact-checking rests primarily on the shoulders of human fact-checkers who laboriously sift through various trustworthy sources, interview subject experts, and check references before reaching a verdict regarding the degree of truthfulness of a factual claim. Compounded with the speed and scale at which misinformation spreads, the demanding process may leave many harmful factual claims unchecked.

In the fight to curb the spread of misinformation, researchers from various disciplines have come forward to assist fact-checkers by creating several automated fact-checking tools and apps. In this dissertation, we focus on studying factual claims and make the following contributions to assist the automated fact-checking efforts:

(1) Understanding a factual claim and parsing the content of the claim to extract its attributes are challenging. We propose a way to represent claims in a structured format to capture various aspects of claims, such as entities involved, their relationships, quantities, points and intervals in time, comparisons, and aggregate structures. We use semantic frames

for the representation of factual claims. We create a set of new semantic frames, a dataset of frame-annotated claims, and a publicly available web-based annotation tool.

(2) To verify a factual claim over a relational database, it is necessary to translate it into a SQL query. However, automatically translating claims to SQL queries is hard. We conduct a preliminary investigative study: (a) to reveal challenges in claim translations and (b) to assess the efficacy of applying a state-of-the-art text-to-SQL parser in translation.

(3) The problem of unchecked claims is exacerbated on social media. We build ClaimPortal, a web-based platform that enables users to monitor, search, and check English factual claims on Twitter. We further demonstrate a semantic-frame-based model to categorize tweets based on the type of factual claims they promote.

(4) One of the critical components in the fact-checking process is automatically assessing the check-worthiness of a piece of information. It is crucial to have a carefully annotated ground-truth dataset that can feed a machine-learning algorithm to predict the check-worthiness of a statement. To bridge this gap, we create a large dataset of claims from all U.S. presidential debates (1960 to 2016) along with the human-annotated check-worthiness label.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

> *Falsehood flies, and truth comes limping after it, so that when men come to be undeceived, it is too late; the jest is over, and the tale hath had its effect: like a man, who hath thought of a good repartee when the discourse is changed, or the company parted; or like a physician, who hath found out an infallible medicine, after the patient is dead.*

Jonathan Swift, *The Examiner No. XIV*, November 9th, 1710

1.1 Motivation

In recent years, the proliferation of misinformation has reached a staggering pace, eroded people's confidence in politics, and has even affected democracies [2]. For example, during the 2016 elections, the misinformation propagated was highly favorable to one side [3]. A recent survey [1] conducted by the Pew Research Center found that 68% of the respondents reported that misinformation hurt their belief in the government; 51% reported that they believed misinformation could impede progress in politics.

Misinformation is not a new challenge. It dates back to the 1890s [2] when journalism and associated newspapers offered little or no well-researched news. However, the novelty of the current misinformation challenge lies in the speed and the scale at which it spreads. Today, misinformation not only undermines people's belief and trust in government institu-

[1] https://pewrsr.ch/37ykPcs

[2] https://www.cits.ucsb.edu/fake-news/brief-history

tions [2], it also costs lives. [3] A recent study has shown that exposure to misinformation has led to hesitancy to get the COVID-19 vaccine [4].

Many tools, practices, and services have emerged in response to the urgent need to fight the dissemination of misinformation. According to a recent report [4] from the Duke Reporters' Lab, as of June 2021, the number of active fact-checking outlets has reached 341, from at least 102 countries. To curb the spread of misinformation, professional fact-checkers and journalists work diligently to debunk falsehoods. The challenge is that they cannot keep up with the amount of falsehoods as fact-checking is time-consuming; verifying one claim typically takes between 4 hours and 1 day [5]. These challenges create an opportunity for automated fact-checking systems.

There has been a considerable response from the academic research communities within computer science, political science, and journalism. These research communities have made significant efforts in studying the spread of misinformation and in aiding fact-checking. Many such efforts led to the development of computational methods and tools in countering misinformation on various fronts, such as identifying claims worth fact-checking from a myriad of sources of digital or traditional media [6, 7, 8, 9, 10], debunking repeated claims by matching them against a collection of already checked claims [11, 12, 13, 14, 15], vetting claims by (1) using supporting, refuting, and related evidence sentences from documents [16, 17, 18, 19], (2) making use of knowledge bases by associating claim entities with knowledge base properties [20, 21, 22, 23, 24, 25, 26], and (3) translating claims into verification queries on relational databases [27, 28, 29]. Several studies aimed at understanding misinformation on several aspects such as its diffusion model [30], correlations between different predictors and an individual's tendency to reject or accept a factual claim [31], the effects of different

[3] https://www.whitehouse.gov/briefing-room/press-briefings/2021/07/15/press-briefing-by-press-secretary-jen-psaki-and-surgeon-general-dr-vivek-h-murthy-july-15-2021/

[4] https://reporterslab.org/tag/fact-checking-database/

corrective strategies on a person's recollection of misinformation and its degradation over time [32], and how influencers wield social media to spread misinformation [33].

Research and development efforts on these fronts can benefit from structured representations of factual claims that capture various aspects of such claims, including the entities involved and their relationships, quantities, points and intervals in time, comparisons, and aggregate structures. With such a modeling capability in place, fact-check assisting tools can exploit the idiosyncrasies of different forms of factual claims. For instance, in translating claims into verification queries over relational databases [27, 29], query templates can be carefully crafted beforehand for different types of claims, and methods can be designed to replace the variables in the query templates with entities and elements from the structured representations. In this dissertation, we analyze factual claims and model them syntactically and semantically with semantic frames [34]. Furthermore, we create a set of semantic frames to represent factual claims, an annotated dataset of factual claims with semantic frames, and a publicly available web-based annotation tool. We also introduce a model that leverages semantic frames in categorizing tweets based on factual claims they promote. We then integrate this model into a web-based platform that we built to monitor, search, and check factual claims on Twitter. We also build a dataset of factual claims for training machine learning models to identify check-worthy claims from the text.

1.2 Dissertation Outline and Contributions

The outline and contributions in this dissertation are as follows.

In Chapter 2, we provide an overview of background information that is relevant. We review fundamentals of the fact-checking process and the related tasks. Furthermore, we discuss existing methods and tasks in assisting fact-checking.

In Chapter 3, we introduce our approach for the structured and semantic modeling of factual claims. We produce a corpus of 20 factual-claim-specific semantic frames, including 11 new frames and nine existing ones from FrameNet [34], all of which are used for representing factual claims. In addition to these frames, we create a dataset of 2,540 fully annotated sentences that can be used to understand the aforementioned frames and to train machine learning models. Additionally, we built an annotation tool to facilitate the annotation of sentences with frame semantics. We discuss possible use cases to leverage factual-claim-specific semantic frames. We then conduct preliminary experiments to assess the efficacy of using factual-claim-specific frames in the claim detection task. Experiments results show that 6 out of 10 of our frames performed better than random selection. ''Occupy rank", "Vote", ''Uniqueness of trait" frames achieved 75%, 74%, and 67% F1 score respectively.

Chapter 4 explains the results of a preliminary investigative study to assess the efficacy of using a state-of-the-art text-to-SQL parser for translating congressional voting-related factual claims into SQL queries. We offer a step-by-step process of creating three small-scale datasets, each with a different writing style: natural language statements (factual claims), questions, and utterances. We compare the parser's performance on the three datasets and find that the parser's performance differs significantly for each dataset. We obtained the best results with the natural language utterance dataset of 70% exact matching accuracy.

In Chapter 5, we lay out the details of two of our fundamental contributions for supporting fact-checking efforts: Firstly, we introduce ClaimPortal, a web-based platform that we built for monitoring, searching, and checking factual claims on Twitter. Here we describe the architecture of ClaimPortal, its components, functions, and the user interface. This section also details our study of categorizing tweets by the type of factual claims they promote using semantic frames. We defined 12 claim categories and a set of mapping semantic frames per category. We then design a method that identifies corresponding

frame(s) for each tweet by utilizing a state-of-the-art frame semantic parser and then maps identified frames to their corresponding claim types (Section 5.1).

Secondly, in Chapter 5, we introduce a dataset of claims from all U.S. general election presidential debates (1960 to 2016) along with the human-annotated check-worthiness label. We describe the preparation process of the dataset, present descriptive statistics of the dataset, propose possible use cases, and explain different fairness policies we have followed while developing the dataset. This dataset is crucial in training machine learning models for detecting factual claims from text (Section 5.2).

In Chapter 6, we summarize our findings.

CHAPTER 2

BACKGROUND

This chapter provides background information to lay out the foundation for the subsequent chapters. We first introduce the fact-checking process that consists of various components requiring computational approaches for automation (Section 2.1). We then give a brief overview of some of the existing projects related to fact-checking (Section 2.1). Finally, we delve into those particular components and summarize current computational approaches (Section 2.1.1 and Section 2.1.2).

2.1 Automation in Fact-checking

The amount of misinformation and the speed at which they spread is way beyond the capacity of current fact-checkers since fact-checking is an intellectually demanding and laborious process. For instance, it takes about one day to research a factual claim and write a typical fact-checking article to vet it [5]. Many harmful claims therefore remain unchecked.

This challenge creates a pressing need for automated fact-checking assistive systems. Fact-checking can be defined as the task of assessing the truthfulness of a claim made in a written or spoken modality. Typically this is a process that mandates several tasks, including claim monitoring, claim spotting, claim matching, claim checking, and verdict presentation. *Claim monitoring* aims to monitor live discourses (e.g., interviews, speeches, and debates), social media, and news to extract content. The task of *claim spotting* aims to detect factual claims that are worth checking. *Claim checking* is the task where research on the veracity of a claim is conducted to reach a verdict on the claim. In *verdict presentation*, a report is created to explain the findings used to reach the verdict.

In recent years academics, journalists, professional fact-checkers, and technology companies brought together their forces to tackle misinformation. As a result of this effort, various apps, services, tools were built to help fact-checkers counter misinformation and disseminate their fact-checks. Here is a brief overview of some of those efforts:

ClaimBuster [1] is the umbrella under which several fact-checking-related projects fall. It 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. For various types of new claims not checked before, ClaimBuster automatically translates them into queries against knowledge bases and reports whether they check out. Furthermore, it provides an API [2] with several models for users to identify check-worthy statements.

The #CoronaVirusFacts Alliance [3] is a group of more than 100 fact-checking organizations from around the world brought together by the International Fact-Checking Network (IFCN) as a joint force to combat misinformation about the coronavirus pandemic. The IFCN created a searchable database [4] that collects all of the falsehoods fact-checked by the #CoronaVirusFacts Alliance. This database contains fact-checks in at least 40 languages from more than 70 countries.

The Tech & Check Cooperative [5] is a project launched by Duke Reporters' Lab to create apps and tools that help fact-checkers do their work and broadcast published fact-checks to new audiences. **(1)** One of the works in this project is an app called **Fact-**

[1] https://idir.uta.edu/claimbuster/

[2] https://idir.uta.edu/claimbuster/api/

[3] https://www.poynter.org/coronavirusfactsalliance/

[4] https://www.poynter.org/ifcn-covid-19-misinformation/

[5] https://reporterslab.org/tech-and-check/

Stream that integrates the work of three important fact-checking organizations, namely The Washington Post, PolitiFact and FactCheck.org. It collects the fact-checked claims in the ClaimReview database and notifies its users of the most recent fact-checks on a continuous basis. **(2) Tech & Check Alerts** is an automated service that sends the participating fact-checkers daily email alerts with the most check-worthy claims they might be interested in verifying. This service uses the ClaimBuster API – an ML-based model that scores claims based on how important it is to vet their truthfulness – on statements from social media posts and official transcripts. **(3) Squash** is a platform that does live fact-checking during political events such as debates. It first converts audio of live event into text, and then searches for matching previously published fact-checks in the ClaimReview database. When matches are found, human editors choose relevant ones and post them on the app. Users therefore see fact-checks of politicians' claims within seconds of their utterance. **(4)** Finally, **ClaimReview** is a tagging system that standardizes the content of fact-checks in a machine-readable way. It enables researchers, journalists and developers to leverage the existing fact-checks in creating new apps and other technologies. ClaimReview is a result of the combined efforts of the Reporters' Lab, Schema.org, and Jigsaw.

These projects play a crucial role in countering misinformation, as fact-checking debunks false claims and deter speakers from making false claims in the future [35].

In the following sections, we provide a brief overview of prior studies on claim spotting and claim checking tasks.

2.1.1 Claim Spotting

Professional fact-checkers are overwhelmed with a large number of claims. They need to minimize the time spent spotting check-worthy claims from large information streams so that they can dedicate their time to vetting the claims. To help achieve this goal, there have been efforts to deploy machine learning models for claim spotting. A claim-

spotting model aims to detect check-worthy claims from such large streams and provide fact-checkers with a ranked list of claims. This ranked list helps fact-checkers prioritize on important claims and avoid negligence. ClaimBuster [6] is the first work that automates the claim spotting task. It trains a machine learning model using human-labeled sentences from past presidential debates. The model produces a score that indicates how likely a sentence contains an important factual claim that should be checked. Over the years, new ClaimBuster models [12, 9, 36] have been developed by expanding training dataset and using deep neural networks. The dataset that was used to build the latest ClaimBuster models is a contribution of this dissertation and the creation of the dataset is explained in detail in Section 5.2. Some models have been developed through participating in shared tasks such as the CLEF CheckThat! Lab [37, 38, 39, 40]. Those claim-spotting models trained on custom CLEF-CheckThat! datasets and they typically use LSTM or transformer-based deep neural networks. Additionally, a number of fact-checking organizations [6] [7] around the world make use of claim spotting models in their fact-checking efforts.

2.1.2 Claim Checking

An automated system for claim checking needs to determine what information is needed to vet a factual claim, search and retrieve supporting information from various sources, such as a knowledge base (KB) or a website, and examine the supporting information in order to reach a verdict [41]. Claim checking approaches can be categorized into four groups based on the data sources used to verify claims.

1) Using Existing Fact-checks

[6] https://fullfact.org/automated

[7] https://team.inria.fr/cedar/contentcheck/

Given a factual claim and a repository of existing fact-checks, the claim matching approach aims to find matching fact-checks for the claim from the repository [15]. In the simplest case, where a claim is identical to one that has been fact-checked, the verdict of the corresponding fact-check can be presented to readers and viewers. However, it is not always as straightforward as finding fact-checks on identical claims. Instead, existing fact-checks are often rephrased. Hence, discovering such fact-checks requires a more general solution. Such a solution can benefit from coreference resolution [42], entity matching [43], paraphrase detection [44], semantic textual similarity [45], and natural language inference [46]. For instance, consider the following two statements: "One in 10 babies born in this country is born in Texas." [8] and "10% of U.S. children are Texans." [9] These two statements are paraphrases of one another, and a paraphrase detection tool or natural language inference tool can be used to reveal that. Thus the verdict on one can effectively help fact-check the other. Shaar et al. [14] created two datasets of input claim - verified claim pairs by obtaining the verified claims from PolitiFact [10] and Snopes. [11]

Another approach uses evidence from both relevant fact-checking articles and related web documents for a given claim. Wang et al. [47] crawled the web and found fact-checking articles based on the ClaimReview markup [12] in web pages. They extracted the fact-checks embedded in such article and discovered a list of relevant, supporting web pages. The resulting repository of fact-checking articles and supporting articles can be matched against a given factual claim. A limitation of this approach is that ClaimReview is not widely

[8] https://www.politifact.com/factchecks/2017/oct/12/joyce-mauk/fort-worth-pediatrician-says-1-10-us-\
born-babies-b/

[9] https://www.politifact.com/factchecks/2020/jan/31/james-white/are-10-children-united-states-texans/

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

[11] https://www.snopes.com/

[12] ClaimReview (https://schema.org/ClaimReview) is a standard schema used by fact-checkers for annotating common structured information, such as claim, claimant and verdict, within fact-checking articles.

adopted by fact-checkers, leaving out many claims and articles from such a repository. A recent study [19] proposed a model to automatically extract structured information from fact-checking articles.

2) Using Web Sources

The second approach employs the Web as a knowledge source to retrieve relevant information that can be used to verify claims. This approach [48, 49, 50] take the credibility of the Web source into consideration to confirm or reject a claim. Some other studies [51, 52, 53, 54, 55] assume a credible source (e.g., Wikipedia articles) is already given and only focuses on retrieving evidence from the source for claim checking. The creation of the FEVER dataset [56] and two shared tasks, FEVER [57] and FEVER 2.0 [58], has stimulated the development of many methods. Most of these methods are comprised of three subtasks: document retrieval, sentence retrieval, and finally using natural language inference to decide the claims' veracity [59]. A study from Nie et al. [60] employs a homogeneous semantic matching network for all the subtasks while the aforementioned ones use different models for each subtask.

3) Using Knowledge Bases (KBs)

Another way to assess the veracity of a claim is to validate it against a knowledge base (KB) or a knowledge graph (KG). Knowledge bases store facts about real-world entities in triples in the form of (head entity, relation, tail entity), e.g., (Microsoft, founded-by, Bill Gates). In recent years, many initiatives have created large-scale knowledge bases, which have become an essential resource for AI-related applications, including fact-checking [25, 26]. Validating a claim using a knowledge base can entail employing natural language processing techniques to convert the claim to a query over the knowledge base.

Another approach is to find triples relevant to the claim and use the connectivity and distance between these relevant triples to assess the claim's truthfulness [61, 20, 21, 22].

4) Using Relational Databases

There are also some efforts to verify numerical claims by translating them into aggregate queries over relational databases. However, compared to all the aforementioned approaches, these efforts are minimal. The lack of claim - SQL query pairs datasets and the complexity in automatically translating claims to SQL queries may have played a role in this. One of the existing works is from Jo et al. [28, 27], where they introduced the AggChecker to translate numerical claims to SQL queries. AggChecker constructs candidate SQL queries based on calculated relevance scores between pre-defined SQL query fragments and the claim keywords given a claim and its associated database. It then uses an expectation-maximization algorithm to compute the probabilities of query candidates. Finally, the AggChecker decides if a claim is likely to be wrong based on its most likely query. Another system proposed by Karagiannis et al. [29] is Scrutinizer. Scrutinizer utilizes four classifiers to extract the fragments of the SQL query from a given claim. The three classifiers work to identify essential elements of each query, such as primary keys values, names of attributes, and relevant relations. The final classifier identifies a generic formula with variables in the place of keys and attribute values. If Scrutinizer cannot predict an element with high confidence, it requires users to build the query.

CHAPTER 3

MODELING FACTUAL CLAIMS WITH SEMANTIC FRAMES *

In this chapter, we present our work on structured and semantic representation of factual claims.

3.1 Introduction

In the development of automated fact-checking systems, researchers can benefit from structured representations of factual claims which capture various aspects of such claims such as the entities involved, their relationships, quantities, points and intervals in time, comparisons, and their aggregate structures. With such a modeling capability in place, fact-check assisting tools can exploit the idiosyncrasies of different forms of factual claims. For instance, in translating claims into verification queries over relational databases [27, 29], query templates can be carefully crafted beforehand for different types of claims, and methods can be designed to replace the variables in the query templates by entities and elements from the structured representations. By modeling factual claims, we can also explore and uncover common semantic structures present in misinformation. An example of this can be seen in a recent study [64] that analyzed pro- and anti-vaccine comments and found that, in both sets of comments, risk-related and causation type words were used more. Such studies could attain greater granularity by identifying semantic structures that correlate

* This chapter is largely adapted from [62]: **Arslan, F.**, Caraballo, J., Jimenez, D., and Li, C. (2020). Modeling Factual Claims with Semantic Frames. *In Proceedings of LREC 2020*.

and also adapted from [63]: **Arslan, F.**, Jimenez, D., Caraballo, J., Zhang, G., and Li, C. (2019). Modeling factual claims by frames. *In Proceedings of the Computation+Journalism Symposium.*

with or represent particular sentence elements, e.g., risk-related or causation type words, through modeling of claims.

Our approach is to adopt and extend the Berkeley FrameNet [1] project, a lexical resource for English built on a theory of meaning called *frame semantics* [34]. This theory "asserts that people understand the meaning of words largely by virtue of the frames which they evoke." [65] In frame semantics, *lexical units* (LUs, i.e., words, phrases, and linguistic patterns) evoke frames. A *frame* describes a type of event, action, situation, or relation, together with *frame elements* (FEs). Frame elements are frame-specific semantic roles that provide additional information to the semantic structure of a sentence.

In this study, we created factual-claim specific frames to represent claims in a structured format. We used fact-checked claims from PolitiFact and analyzed their internal structures. We grouped the claims sharing common syntactic and semantic patterns in order to form conceptual categories of claims that convey similar meanings. This process yielded a total of 20 claim categories. For each claim category, we identified all possible terms (words, phrases, and linguistic patterns) specific to the category that can become lexical units of frames. We mapped each of the identified terms to the LUs of frames in FrameNet so as to identify existing frames that represent our claim categories. For the claim categories where we found a matching frame, we used that frame to model factual claims belonging to the category. For the remaining claim categories, we created new frames. As a result, we identified 9 matching frames and created 11 new frames. For each new frame, we provide its frame definition, a set of associated FEs along with their descriptions, a set of LUs, annotated example sentences, and frame-to-frame relations. Figure 3.1 shows a new frame "Vote" created for characterizing claims about someone's voting decision towards an issue. "Agent" and "Issue" are two of the frame elements. "Agent", a conscious entity, holds a

[1] https://framenet.icsi.berkeley.edu/

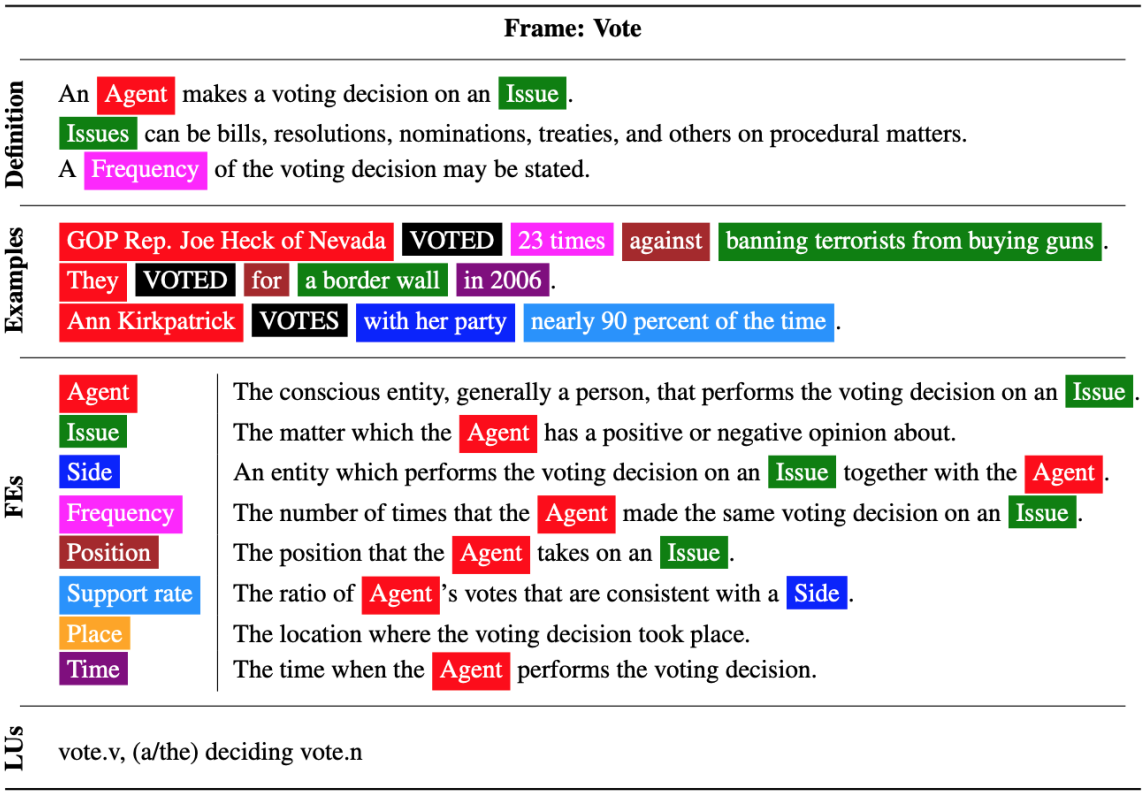


Figure 3.1: The *Vote* frame, one of the new factual-claim specific frames

positive or negative opinion about an "Issue" and votes on it. The lexical units of the "Vote" frame are "vote" and "(a/the) deciding vote" in the verb and noun forms, respectively.

To support further studies that leverage the outcome of this work, we created a corpus of claims fully annotated with the aforementioned 20 factual-claim specific frames. We used $4,664$ fact-checks from the "Share the Facts" database [2] that is regularly updated by several fact-checking organizations. Since some of these factual claims consist of multiple sentences, we split the claims into sentences. The corpus size thus became $6,017$ individual sentences. For each lexical unit belonging to one of the 20 frames, we identified sentences containing these LUs and further annotated these sentences with their respective frame elements. A

[2] http://www.sharethefacts.org/

total of $2,540$ sentences, each associated with one or more frames, were annotated using the 20 frames. This resulted in $3,616$ frame annotations for the $2,540$ sentences.

This chapter describes in detail our work on modeling factual claims using frame semantics—the first such study to the best of our knowledge. Our dataset of frame definition files and annotated sentences for the aforementioned 20 factual-claim specific frames are publicly accessible and permanently archived at https://zenodo.org/record/3710507. We also built a public web-based frame annotation tool FrameAnnotator, [3] to aid annotating sentences. FrameAnnotator supports full-text annotation and encodes annotated sentences in the same XML format used in FrameNet. These resources enable other researchers to make their own local extensions to FrameNet.

3.2 Modeling Factual Claims

3.2.1 Claim Modeling Process

To model factual claims, we began with a collection of 3,643 fact-checks sourced from PolitiFact. We sampled 969 claims that were representative of the entire dataset. The steps in the process of factual claim modeling are explained below. All of these steps were manually conducted.

Analyzing claims: We grouped the sampled claims by common syntactic and semantic patterns that they shared and avoided the creation of numerous groups each with only a few claims in it. This resulted in a set of 20 conceptualized claim categories. A claim can express multiple meanings, and hence can belong to various claim categories.

Identifying category specific terms: The process explained in this step was applied to each of the 20 claim categories generated in the previous step. For each claim category, we identified all possible terms (words, phrases, and linguistic patterns) specific to the category.

[3] https://idir.uta.edu/frameannotator/

We then enhanced the list of identified terms by including their related words. For instance, one of our claim categories, "Oppose and Support", is about an individual supporting or opposing an issue. The list of words that we identified for this claim category includes verbs "support", "oppose", and "back", prepositions "for" and "against", and nouns "supporter" and "opponent". We then expanded the list with the words closely related to the ones in the list. For instance, we added prepositions "in favor of" and "pro" to the list as they are closely related to the previous words. These identified terms are potential candidates for lexical units of the frame corresponding to the claim category. Identifying lexical units is an iterative process as the list can be expanded later with additional words.

Reusing FrameNet frames: We used the following process to identify FrameNet frames that represent some of the 20 claim categories. For each claim category specific term, we identified all the corresponding lexical units that were present in FrameNet. This was followed by identifying all the frames evoked by these lexical units. We then analyzed all the identified frames to select the most frequently evoked frame. For instance, Table 3.1 shows the terms identified for one category and the corresponding FrameNet frames associated with each term. The most frequent frame is "Taking sides". This process resulted in the identification of 9 FrameNet frames (shown in Section 3.2.2.2) that matched our claim categories. Identifying the frames matching 9 out of 20 claim categories shows indirect evidence of the robustness of our claim category creation process.

Creating new frames: We created 11 new frames for the remaining claim categories. We used the terms identified in the previous step as the lexical units of the new frames. We then manually identified frame elements for each of the 11 frames from the subset of sentences belonging to those frames. We further described each of these frame elements (FEs) based on their role in their frame. We then annotated sentences from the subset we used for each

Table 3.1: Terms from the Oppose and Support category and their corresponding FrameNet frames

Term	FrameNet Frames
against.prep	Special contact, Taking sides
back.v	Funding, Self motion, Taking sides
for.prep	Duration relation, Taking sides
in favor.prep	Taking sides
support.v	Evidence, Funding, Supporting, Taking sides
supporter.n	Taking sides
opponent.n	Taking sides
oppose.v	Taking sides
pro.adv	Taking sides

frame according to the generated FEs. Finally, we wrote a definition for each of the new frames.

3.2.2 A Corpus of Factual-Claim Specific Frames

The outcome of this work resulted in 20 factual-claim specific frames, 171 frame elements (FEs), and 294 lexical units (LUs). Eleven of those frames along with 50 FEs and 27 LUs were newly created. Table 3.2 shows the distribution of FEs and LUs among these frames. The "Statement" frame contains the most FEs (20) and the most LUs (79). All the frames with at least 10 FEs are the existing frames from FrameNet. The 9 frames we leveraged from FrameNet are listed in section 3.2.2.2. In the following sections, we briefly describe each frame and provide two sample annotated sentences with lexical units in boldface and frame elements in square brackets.

Table 3.2: Descriptive statistics of the corpus of factual-claim specific frames

Frame	# of FEs	# of LUs
Causation	12	39
Cause change of position on a scale	15	26
Change position on a scale	25	56
Capability	10	17
Comparing entities	5	1
Comparing at two different points in time	6	1
Conditional occurrence	3	8
Correlation	2	2
Creating	19	11
Occupy rank	5	3
Occupy rank via ordinal numbers	5	11
Occupy rank via superlatives	5	12
Ration	4	2
Recurring action	4	1
Recurring action in frequency	4	1
Statement	20	79
Taking sides	12	18
Taking sides consistency	4	3
Uniqueness of trait	3	1
Vote	8	2

3.2.2.1 New Frames

1. Taking sides consistency. This frame is about the consistency of an *Agent's Stance* towards an *Issue*. The *Agent* either alters or maintains his/her *Stance*. The *Stance* may not be explicitly stated.

[Republicans Chuck Grassley, John Boehner and John Mica $_{AGENT}$] **flip-flopped** [on providing end-of-life counseling for the elderly $_{ISSUE}$].

[Donald Trump $_{AGENT}$] has **changed** [his mind $_{STANCE}$] [on abortion $_{ISSUE}$].

2. Recurring action. This frame describes a repetitive *Action* that is performed by an *Agent* at the interval of a *Time_span*.

[Last year $_{TIME}$], [Exxon $_{AGENT}$] [pocketed nearly $4.7 million $_{ACTION}$] **every** [hour $_{TIME\_SPAN}$].

[Undocumented immigrants $_{AGENT}$] [pay $12 billion of taxes $_{ACTION}$] **every** [single year $_{TIME\_SPAN}$].

3. Recurring action in frequency. This frame is about a repetitive *Action* that is performed by an *Agent* at a given *Frequency*.

[Chemical weapons have been used $_{ACTION}$] probably [20 $_{FREQUENCY}$] **times** [since the Persian Gulf War $_{TIME}$].

[Trump $_{AGENT}$] [has taken business bankruptcies $_{ACTION}$] [six $_{FREQUENCY}$] **times**.

4. Vote. An *Agent* makes a voting decision on an *Issue*. *Issues* can be bills, resolutions, nominations, and treaties, and others on procedural matters. A *Frequency* of the voting decision may be stated.

[Mitch McConnell $_{AGENT}$] **voted** [three times $_{FREQUENCY}$] [for $_{POSITION}$] [corporate tax breaks that send Kentucky jobs overseas $_{ISSUE}$].

[In the Senate $_{PLACE}$], [Mike DeWine $_{AGENT}$] **voted** [with Hillary Clinton $_{SIDE}$] [to let illegal immigrants receive Social Security $_{ISSUE}$].

5. Correlation. This frame shows the connection or relationship between the occurrences of *Event_1* and *Event_2*.

Whenever [we raise the capital gains tax $_{EVENT\_1}$], [the economy has been damaged $_{EVENT\_2}$].

Every time [we've cut the capital gains tax $_{EVENT\_1}$], [the economy has grown $_{EVENT\_2}$].

6. Comparing two entities. This frame is about comparing two entities using a *Comparison_criterion* while qualifying with a *Degree*.

[Hillary Clinton $_{ENTITY\_1}$] [has been in office and in government longer $_{COMPARISON\_CRITERION}$] **than** [anybody else running here tonight $_{ENTITY\_2}$].

[African-American children $_{ENTITY\_1}$] are [500 percent more $_{DEGREE}$] [likely to die from asthma $_{COMPARISON\_CRITERION}$] **than** [white kids $_{ENTITY\_2}$].

7. Comparing at two different points in time. This frame is about comparing an *Entity* with itself at two different points in time using a *Comparison_criterion* while qualifying with a *Degree*.

[The average family $_{ENTITY}$] is [now $_{FIRST\_TIME\_POINT}$] [bringing home $4,000 less $_{COMPARISON\_CRITERION}$] **than** they did [just five years ago $_{SECOND\_TIME\_POINT}$].

[More $_{DEGREE}$] [private-sector jobs $_{ENTITY}$] [were created $_{COMPARISON\_CRITERION}$] [in the second year of the Obama administration $_{FIRST\_TIME\_POINT}$] **than** [in the eight years of the Bush administration $_{SECOND\_TIME\_POINT}$].

8. Occupy rank via ordinal numbers. This frame is about an *Item* in the state of occupying a certain *Rank* specified by an ordinal number within a *Comparison_set*.

[The United States $_{ITEM}$] is [**65th** $_{RANK}$] [out of 142 nations and other territories $_{COMPARISON\_SET}$] [on equal pay $_{DIMENSION}$].

[In the mid 1990s $_{TIME}$], [Florida $_{ITEM}$] was [**no. 1** $_{RANK}$] [in violent crime $_{DIMENSION}$] [in America $_{COMPARISON\_SET}$].

9. Occupy rank via superlatives. This frame is about an *Item* in the state of occupying a certain *Rank* specified by a superlative within a *Comparison_set*.

[Job growth in the United States $_{ITEM}$] is [now $_{TIME}$] at [the **fastest** $_{RANK}$] [pace $_{DIMENSION}$] [in this country's history $_{COMPARISON\_SET}$].

[The U.S. $_{ITEM}$] is [the **largest** $_{RANK}$] [energy producer $_{DIMENSION}$] [in the world $_{COMPARISON\_SET}$].

10. Ratio. In this frame, a *Criterion* determines a *Ratio* that quantifies the size of the subset of a larger *Group*.

[More than 72 $_{RATIO}$] **percent of** [children in the African-American community $_{GROUP}$] are [born out of wedlock $_{CRITERION}$].

[Since the recession ended $_{TIME}$], [about 85 $_{RATIO}$] **percent of** [income growth $_{GROUP}$] [went to the top 1 percent $_{CRITERION}$].

11. Uniqueness of trait. This frame distinguishes a *Unique entity* from a *Generic entity* based on a specific *Trait* where a *Trait* is some property, quality, point-of-view, or an arbitrary construct which is generally understood to be an attribute of an entity.

[The United States $_{UNIQUE\_ENTITY}$] is **the only** [advanced country on Earth $_{GENERIC\_ENTITY}$] [that doesn't guarantee paid maternity leave to our workers $_{TRAIT}$].

[Florida $_{UNIQUE\_ENTITY}$] is now **the only** [state in the nation $_{GENERIC\_ENTITY}$] [to tax commercial leases $_{TRAIT}$].

3.2.2.2 Existing FrameNet Frames

The following frames are adapted from the Berkeley FrameNet corpus. We provide a sample annotated sentence with lexical units in boldface and frame elements in square brackets. The sample sentences are from our corpus of factual-claim specific frames.

1. Takins sides: "A *Cognizer* has a relatively fixed positive or negative point of view towards an *Issue*." [4]

[Hillary Clinton $_{COGNIZER}$] **supported** [North American Free Trade Agreement $_{ISSUE}$].

[I $_{COGNIZER}$] have [consistently $_{DEGREE}$] **opposed** [shutdowns $_{ISSUE}$].

2. Statement: "This frame contains verbs and nouns that communicate the act of a *Speaker* to address a *Message* to some *Addressee* using language." [5]

[Yesterday $_{TIME}$] [for the first time $_{FREQUENCY}$] [she $_{SPEAKER}$] **said** [she wants to renegotiate trade agreements $_{MESSAGE}$].

[Ronald Reagan $_{SPEAKER}$] **talked** [about converting the United States to the metric $_{TOPIC}$].

3. Causation: A *Cause* leads to an *Effect*. "Alternatively, an *Actor*, a participant of a (implicit) *Cause*, may stand in for the *Cause*." [6]

[Global warming $_{CAUSE}$] would **cause** [sea levels $_{AFFECTED}$] [to rise on average not one yard but many yards $_{EFFECT}$] [in as soon as 50 years $_{TIME}$].

---

[4] https://framenet2.icsi.berkeley.edu/fnReports/data/frame/Taking_sides.xml

[5] https://framenet2.icsi.berkeley.edu/fnReports/data/frame/Statement.xml

[6] https://framenet2.icsi.berkeley.edu/fnReports/data/frame/Causation.xml

Due to [actions by President Barack Obama $_{CAUSE}$] [the Burger King national headquarters announced this month that they will be pulling their franchises from our military $_{EFFECT}$].

4. Capability: "An *Entity* meets the pre-conditions for participating in an *Event*." [7]

[Former President George W. Bush and former Vice President Dick Cheney $_{ENTITY}$] are **unable** [to visit Europe $_{EVENT}$] [due to outstanding warrants $_{CIRCUMSTANCES}$].

[If someone is known or suspected as a terrorist $_{CIRCUMSTANCES}$], [they $_{ENTITY}$] **can** not [just walk in and buy a firearm $_{EVENT}$].

5. Cause change of position on a scale: "This frame consists of words that indicate that an *Agent* or a *Cause* affects the position of an *Item* on some scale (the *Attribute*) to change it from an initial value (*Value_1*) to an end value (*Value_2*)." [8]

[In the last two years $_{TIME}$], [we $_{AGENT}$] have **reduced** [the deficit $_{ATTRIBUTE}$] [by $2.5 $_{DIFFERENCE}$].

[Since the fourth quarter of last year $_{TIME}$] [the U.S. economy $_{CAUSE}$] has **added** [almost 50,000 $_{DIFFERENCE}$] [jobs $_{ITEM}$] [in the coal sector $_{PLACE}$].

6. Change position on a scale: "This frame consists of words that indicate the change of an *Item's* position on a scale (the *Attribute*) from a starting point (*Initial_value*) to an end

[7] https://framenet2.icsi.berkeley.edu/fnReports/data/frame/Capability.xml

[8] https://framenet2.icsi.berkeley.edu/fnReports/data/frame/Cause_change_of_position_on_a_ scale.xml

point (*Final_value*). " [9]

[Since 2007 $_{TIME}$], [Texas $_{ITEM}$] has **gained** [440,000 people $_{DIFFERENCE}$] while Maryland has lost 20,000.

[DUI arrests $_{ITEM}$] **dropped** [significantly $_{DEGREE}$] [in Tampa $_{PLACE}$] [once Uber began operating here $_{CIRCUMSTANCES}$].

7. Creating: "A *Cause* leads to the formation of a *Created_entity*." [10]

[In the last 29 months $_{TIME}$], [our economy $_{CREATOR}$] has **produced** [about 4.5 million private-sector jobs $_{CREATED\_ENTITY}$].

[During President Barack Obama's tenure $_{TIME}$], [the United States $_{CREATOR}$] has **created** [15 million new jobs $_{CREATED\_ENTITY}$].

8. Occupy rank: "This frame is about *Items* in the state of occupying a certain *Rank* within a hierarchy." [11]

[The U.S. $_{ITEM}$] only **ranks** [25th $_{RANK}$] [worldwide $_{COMPARISON\_SET}$] [on defense spending as a percentage of GDP $_{DIMENSION}$].

---

9 https://framenet2.icsi.berkeley.edu/fnReports/data/frame/Change_position_on_a_scale.xml

10 https://framenet2.icsi.berkeley.edu/fnReports/data/frame/Creating.xml

11 https://framenet2.icsi.berkeley.edu/fnReports/data/frame/Occupy_rank.xml

The following statement contains two occupy rank frame instances. Thus, we present both annotations as follow:

[Illinois $_{ITEM}$] is **ranked** [number one $_{RANK}$] [in the nation $_{COMPARISON\_SET}$] [for a decline in treatment capacity $_{DIMENSION}$] [between 2007 and 2012 $_{TIME}$], and is now ranked the third worst in the country for state-funded treatment capacity.

[Illinois $_{ITEM}$] is ranked number one in the nation for a decline in treatment capacity between 2007 and 2012, and is [now $_{TIME}$] **ranked** [the third worst $_{RANK}$] [in the country $_{COMPARISON\_SET}$] [for state-funded treatment capacity $_{DIMENSION}$].

9. Conditional occurrence: "A *Consequence* is presented as occurring if the *Profiled_possibility* occurs." [12]

[We would create thousands of jobs in Colorado $_{CONSEQUENCE}$], **if** [the Keystone Pipeline were to be built $_{PROFILED\_POSSIBILITY}$].

[You can absolutely get a gun if you have several felonies $_{CONSEQUENCE}$] **as long as** [you buy it on the Internet or at a gun show $_{PROFILED\_POSSIBILITY}$].

3.3 Annotation

This section discusses the source of the annotated sentences, the annotation process, the annotation tool that we created to assist this process, and the statistics of the annotated sentences.

[12] https://framenet2.icsi.berkeley.edu/fnReports/data/frame/Conditional_occurrence.xml

Data Source: In order to construct a sizable corpus of annotated sentences, we used fact-checked claims from the "Share the Facts" database. The "Share the Facts" database contains fact-checks annotated with the ClaimReview schema–a schema.org standard which specifies a standardized format for fact-checks. The initial dataset had around 18,000 fact-checks compiled from 34 different fact-checking organizations.[13] The distribution of fact-checks from the top contributors is as follows: Gossip Cop (9082),[14] PolitiFact (4644), the Washington Post (3100),[15] FactCheck.org (928),[16] and Snopes (31). We removed redundant fact-checks and irrelevant fact-checks, such as those from international organizations and those associated with Hollywood gossip magazine sections. After cleaning up the dataset and splitting fact-checks with multiple sentences, we ended up with 6,017 fact-checks that we deem to be of high-quality with respect to our task.

Annotation Process: For each lexical unit in our corpus, we gathered all the sentences containing the lexical unit from the preprocessed "Share the Facts" dataset. We manually filtered out sentences that did not use a given lexical unit in the same context as the other sentences that they were initially grouped with via the rudimentary gathering step. We denote this group of sentences that share a lexical unit as S. We took sentences from S and marked syntactic elements in each sentence that corresponded to the frame elements for a given frame.

Annotation Tool: We built a public web-based frame annotation tool [3] in order to facilitate annotating sentences with frame semantics. We now use Figure 3.2 to explain how to use the annotation tool. First, a user uploads sentences in region 2 and selects a frame from region 3. Once sentences have been loaded, they appear in region 5. Here, a user can annotate a

[13] We downloaded the dataset on August 27, 2018. The dataset now contains more than 20,000 fact-checks.

[14] https://www.gossipcop.com/

[15] https://www.washingtonpost.com/

[16] https://www.factcheck.org/

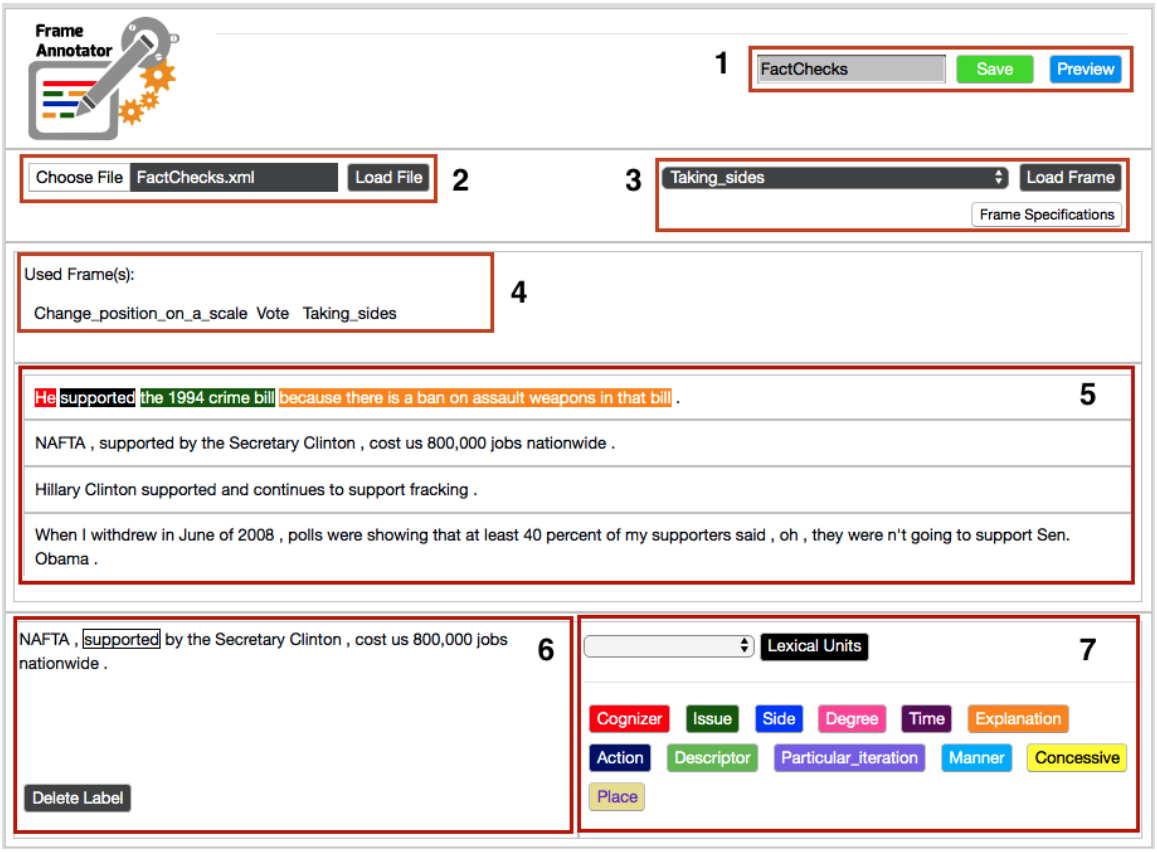


Figure 3.2: The user interface of *FrameAnnotator*

sentence by simply clicking on it and sending it to region 6. When a sentence is populated in region 6, the tool attempts to highlight a potential lexical unit. An example of this can be seen in Figure 3.2. An annotator can click and drag over a sentence fragment to select it in region 6. When a sentence fragment is selected, the tool highlights the region to provide feedback to the annotator that the system is aware of their selection. Once the fragment is highlighted, it may be marked by selecting the appropriate frame element in region 7. When frame elements are marked, their respective frames appear in region 4 to inform the annotator on what frames have been used. To save progress, an annotator may assign a filename and click the "Save" button in region 1. The exported annotations are stored in

XML format, enabling programs to consume the annotations or allowing annotators to pick up where they left off.

Annotation Statistics: As mentioned earlier, our efforts led to a total of $2,540$ fully annotated sentences with $3,616$ frame annotations. Figure 3.3 shows a fully annotated sentence. Most sentences, $1,955$, from the set of $2,540$ have only 1 associated frame and 478 had two frame instances. The rest of the sentences had between 3 and 10 frame instances, with the number of sentences decreasing while the number of associated frames increases. Figure 3.4 shows a breakdown of the number and percentage of annotated sentences in the corpus by individual frames. Information about the composition of each frame instance is presented in Appendix A.

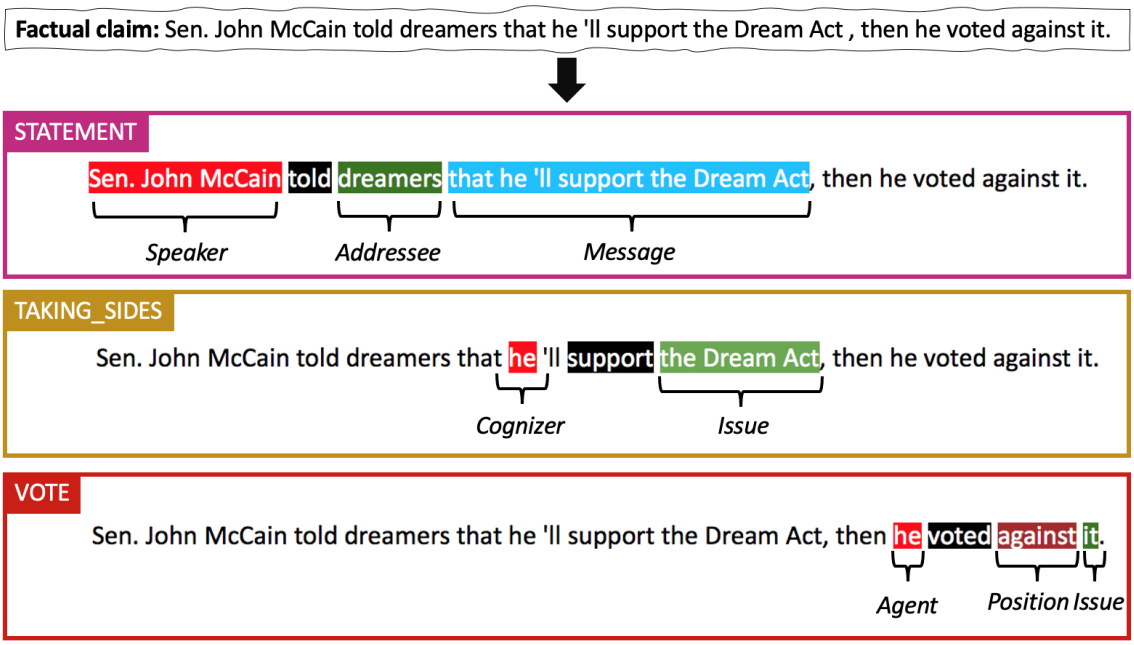


Figure 3.3: A fully-annotated example factual claim is shown along with its corresponding semantic frames. Each frame indicates a factual claim type and shows color-coded textual spans as frame elements to present the properties of the claim.

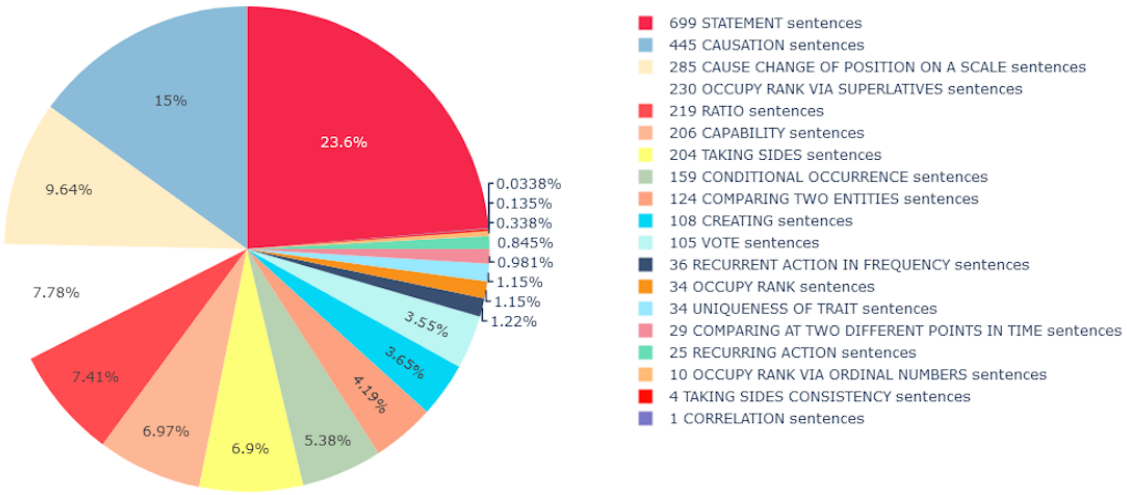


Figure 3.4: Distribution of annotated sentences by frame instance over corpus

3.4 Potential Uses of the Corpus of Factual-Claim Specific Frames

In this section, we outline some key research areas where the corpus of the factual-claim specific frames can be used.

Potential Uses in Fact-checking: To understand where we can leverage the corpus of factual-claim specific frames in the fact-checking process, we briefly introduce that process for FactCheck.org, a major fact-checking organization.[17] First, journalists identify "statements of fact" made by people of interest in various forums. They then research the identified factual claims by considering the speakers' supporting information and various primary sources. Once the independent research is synthesized into a story, that story goes through a rigorous editing process to ensure quality and veracity. This process can be improved by the application of our work in three areas, including 1) identification of "statements of fact", 2) avoiding duplication of work by "matching" repeated "statements of fact" to their corresponding existing fact-checks, and 3) translating them to structured queries that can be verified over a reliable knowledge base.

[17] https://www.factcheck.org/our-process/

Other Potential Use Cases: Outside of automated fact-checking, this work has potential use in the areas of browsing and search, the academic study of factual claims and natural language processing tasks.

A recent paper described a browsing and search system for tweets that may contain factual claims [66]. The claim categorization feature of this system leverages our work to train the model responsible for categorizing the factual claims present in tweets. Other systems that aim to add a faceted search interface for users to browse/search for certain types of natural language text may benefit from additional labeled data to train their natural language models on.

Factual claims are studied academically in fields such as social science, journalism and computer science. Some studies, such as [64], analyze the distributions of word tokens across different corpora and derive insights from these distributions—our work may enable the analysis of frame and frame element distributions. This could lead to new findings derived from semantic similarities between corpora as opposed to syntactic similarities.

Many natural language processing tasks can potentially benefit from the usage of frames, including question answering [67], information extraction [68], sentiment analysis [69], and machine translation [70]. Our corpus may be leveraged in these tasks as an additional source of data for models to learn from.

3.4.1 Claim Spotting

Claim spotting is a necessary task in the fact-checking process to identify claims worthy of fact-checking from natural language sentences. The task not only consists of identifying claims but also prioritizing them for fact-checking. In recent years, a significant amount of research efforts have been dedicated to the development of claim spotting models. Early models relied on supervised classifiers such as SVM or logistic regression trained on hand-engineered features [6, 7, 8]. In contrast, recent approaches utilize neural sentence

embeddings [71, 9, 10, 36]. A number of fact-checking organizations [18] [19] [20] around the world make use of claim spotting models in their fact-checking efforts to detect claims to check. Claim spotting is one particular task that can benefit from claim specific frames. With factual frames in hand, we can remodel the claim spotting task as identifying and prioritizing claims that have been found to be affiliated with at least one of the 20 frames. In Section 3.4.1.1 we demonstrate the use of semantic frames in claim spotting.

3.4.1.1 Preliminary Experiments

We conducted a preliminary experiment to assess the efficacy of our study. We used open-sesame [72], an open-source frame semantic parser for automatically identifying FrameNet frames and their frame-elements from sentences. Open-sesame, a syntax-free system, is built on softmax-margin segmental recurrent neural nets and executes an array of tasks: target identification, frame identification, and argument identification. Target identification is the identification of the words or expressions that evoke the frames. Frame identification is identifying the frame that each target evokes. Argument identification is the identification of the frame elements and their corresponding span of text for each of the frames that a sentence triggers.

We utilized open-sesame for a claim detection task, more specifically to identify the frames that were evoked by a factual claim under consideration. We added three new frames along with their lexical units and hand-engineered annotated sentences into the FrameNet 1.7 dataset. The inserted frames are the Vote, Uniqueness of trait, and Recurring action frames. We chose these frames as they have a large enough number of sentences associated with them as to allow a satisfactory training phase.

[18] https://fullfact.org/automated

[19] https://team.inria.fr/cedar/contentcheck/

[20] https://reporterslab.org/tech-and-check/

Table 3.3: Frame prediction performance, in terms of Precision (P), Recall (R) and F-measure (F_1). Where avg_w denotes the weighted average of corresponding measure across ten frames. The number in parentheses following the frame name is the number of sentences used for evaluating that frame.

	P	R	F_1
Cause change of position on a scale (156)	0.73	0.60	0.66
Capability (48)	0.32	0.79	0.46
Causation (256)	0.41	0.48	0.44
Creating (114)	0.92	0.31	0.46
Change position on a scale (167)	0.45	0.74	0.56
Occupy rank (35)	0.82	0.69	0.75
Recurring action (29)	0.23	0.66	0.34
Taking sides (129)	0.52	0.46	0.49
Uniqueness of trait (33)	0.54	0.88	0.67
Vote (104)	0.61	0.94	0.74
avg_w	0.56	0.60	0.54

We retrained open-sesame on this extended FrameNet 1.7 dataset and followed this by an evaluation of the trained model with the help of the "Share the Facts" dataset. [21] Since this dataset included some claims that were irrelevant for our current task of political fact-checking, we removed any fact-checks from international organizations and those associated with Hollywood gossip magazine sections. We evaluated open-sesame's frame identification performance for the three new frames (i.e., the "Vote", "Uniqueness of trait", and "Recurring action" frames) in addition to the seven pre-existing FrameNet frames (i.e., the "Taking

[21] The "Share the Facts" dataset is the result of a joint effort by several prominent fact-checking organizations that aims to create a standardized format of fact-checks. The dataset now contains around 20,000 fact-checks and counting. (http://www.sharethefacts.org/)

sides", "Occupy rank", "Creating", "Capability", "Causation", "Change position on a scale", and "Cause change of position on a scale" frames).

3.4.1.2 Results and Discussion

Table 2 depicts the performance results for each of these frames. From the results we see that the Vote and Uniqueness of trait frames performed in line with the other pre-established frames. The recurring action had a low precision and thus lower F1-score. We also see some low scores for some of the pre-established frames, but we expect that as we are able to create a more robust labeling process we will have more data to feed the neural network to improve its performance. We can also look at fine tuning frame elements and or lexical units from what they are currently defined as. However, the latter should only be necessary if we do not see a noticeable improvement with the inclusion of more training data. It should also be noted that during training, we were not only training the model to detect these frames but the entirety of the FrameNet frames. Thus another possible option would be to train the model only on the 20 frames we will eventually focus on to see if that improves performance and produces sound results. Overall, for preliminary results, we are satisfied in seeing that two of our frames performed decently as that validates the direction we are heading in.

3.4.2 Claim Matching

Given a new factual claim, claim matching is the process of partially or fully matching the claim with supporting or refuting fact-checked claims stored in a repository. In the best-case scenario, a new factual claim is a perfect match with an existing factual claim and a user can be provided with the verdict of the claim's veracity. In other scenarios, we can still leverage fact-checked claims, particularly so when the new claim is partially supported or refuted by existing fact-checked claims.

1. [GOP Rep. Joe Heck of Nevada $_{AGENT}$] **voted** [23 times $_{FREQUENCY}$] [against $_{POSITION}$] [banning terrorists from buying guns. $_{ISSUE}$]
2. [Heck $_{AGENT}$] **voted** [nay $_{POSITION}$] [on tighter gun-control laws. $_{ISSUE}$]
3. [Heck $_{AGENT}$] **voted** [for $_{POSITION}$] [stronger gun-control. $_{ISSUE}$]

Figure 3.5: A fact-checked claim with similar and opposite factual claims.

The modeling done in this chapter can help us address the claim matching task by comparing the properties of the claims (i.e. entities, quantities, time intervals, etc.) presented in the frame elements of each claim. The similarity or difference in the corresponding frame elements for each claim can be presented to the user to conclude whether the new factual claim is partially similar to or opposite of the previously fact-checked claim. An example of one comparison between similar and opposite claims is provided in Figure 3.5. The claims in Figure 3.5 are talking about the same individual and similar issues. Matching the new claims to the fact-checked claim (the first claim in the figure) could provide insight to a user about the veracity of the new claims.

3.4.3 Claim to Query Translation

Claim-to-query translation is the process of mapping a given input claim to a structured query that can be run on a knowledge base to verify the given claim. The mapping process is not straightforward as it requires understanding: what is being asked, what context it is being asked in, and identifying any key entities and their relationships in order to answer the specific question that is implicitly introduced by the claim. While there are approaches to entity matching, and relationship matching (e.g., SpaCy,[22] TextRazor,[23] etc.) there is still work to be done to correctly map these elements to a structured query that can be applied on

[22] https://spacy.io/

[23] https://www.textrazor.com/

a knowledge base. Currently some industrial solutions seem to do this behind the scenes (e.g., Wolfram Alpha[24]). However, these are black-box systems and thus open-source and robust solutions still need to be developed.

In the context of claim-to-query translation, frames can be used to identify the key elements in a claim and how they relate to each other within the context of a given frame. This then enables researchers to create query templates that can directly make use of the parsed frame elements extracted from the claim. These query templates can be general to some extent as they can directly relate to a particular frame. Another possibility would be to have a few query templates per frame depending on how complex of a structure the frame is able to represent. One frame that particularly lends itself to this process is the "Vote" frame. It is easy to envision using public voting records to create a knowledge base and then creating one or a few query templates that can make use of the frame elements (e.g., agent, issue, side, frequency, etc.) from the "Vote" frame in order to verify claims of this nature.

[24] https://www.wolframalpha.com/

CHAPTER 4

VERIFYING CONGRESSIONAL VOTING STATEMENTS: A PRELIMINARY INVESTIGATIVE STUDY

4.1 Introduction

Over the past few years, as social media sites have become major channels for the quick dissemination of news, the speed and scale at which misinformation spreads have become a significant threat to tackle across the globe. Falsehoods can impact our society severely in many aspects, such as eroding people's confidence in government institutions, causing uncertainty and mental disturbance (e.g., the ongoing COVID-19 infodemic [73]), creating vaccine hesitancy [4], and even costing people's lives. [1] Computational fact-checking has been proposed to support journalists and fact-checkers with automatic verification of factual claims.

Fact-checking is the task of evaluating the truthfulness of a factual claim. A fact-checking system must determine what information is needed to vet a factual claim, retrieve the information from various sources, such as a knowledge base (KB) or relational database, and then evaluate the data to assign a verdict [41].

Relational databases store a vast amount of data in a structured format and provide an efficient and flexible way to access it. However, accessing relational databases requires users to have a working knowledge of query languages such as SQL and the underlying database schema. This challenge has been the impetus for the development of text-to-SQL systems as a promising area of research during the past few decades. The recent advances in deep

[1] https://www.whitehouse.gov/briefing-room/press-briefings/2021/07/15/press-briefing-by-press-secretary-jen-psaki-and-surgeon-general-dr-vivek-h-murthy-july-15-2021/

learning networks [74] and pretrained language models [75, 76, 77] have fueled the interest in this area. Additionally, with the creation of two new large-scale datasets–WikiSQL [78] and SPIDER [79]–several new text-to-SQL models [80, 81, 82, 83, 84, 85, 86] have been introduced recently.

Existing text-to-SQL models are trained for natural language utterances, where the language is quite different from the language of factual claims, which are grammatically complete statements. Factual claims are natively more ambiguous than natural language utterances. Let us consider the following voting-related factual claim and the corresponding natural language utterance that need to be verified on the congressional voting database:

Factual claim: "Sen. John McCain told dreamers that he'll support the Dream Act, then he voted against it."

Natural language utterance: "Show all information regarding votes whose descriptions contain 'Dream Act' and John McCain voted on."

The natural language utterance here directly expresses the query requirements, while the factual claim is composed of multiple claims and multiple sentence parts, not all of which are related to voting. Thus, a text-to-SQL system needs to pick the claim related to voting and do co-reference resolution for "he" and "it" for a correct query translation.

Existing state-of-the-art text-to-SQL models use deep-learning networks, especially transformer-based neural networks. However, training a deep learning model is not just costly in terms of computing resources and time but also requires large amounts of data in order to produce a good performing model. Training a deep learning model for translating factual claims to SQL queries is not currently feasible primarily due to the lack of a dataset of claims and corresponding SQL. Therefore, in this study, we leveraged a state-of-the-art text-to-SQL model, SmBop [86], to assess the efficacy of such a model in generating a SQL query for a given claim and database.

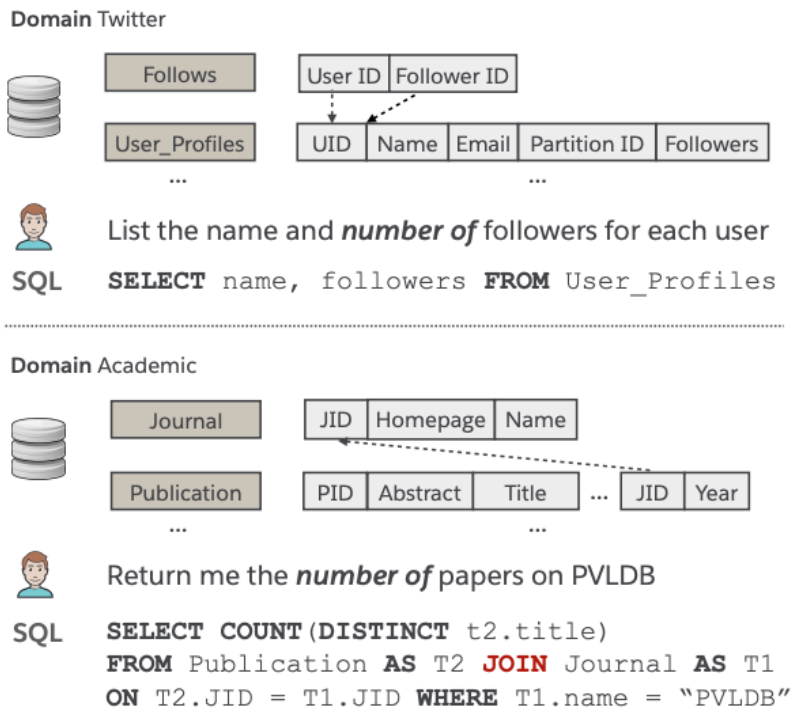


Figure 4.1: Two questions from Spider dataset (the figure is taken from [1])

In order to evaluate the performance of SmBop, we generated a small dataset of 65 factual claims and SQL pairs. SmBop is trained on the Spider dataset, which is a large-scale complex and cross-domain dataset. Figure 4.1 shows two examples of the question, SQL query pairs from the Spider dataset. We can see that both questions directly expose the intent of the questions. Thus, it is much easier for a text-to-SQL parser to generate an accurate SQL query for such a question. However, textual statements, such as a factual claims, are ambiguous and require a parser to make inferences to understand the intent of the statement for generating its corresponding SQL query. For this reason, we generated two more datasets, where one of them contains automatically translated questions from the claims, and the other has manually mimicked questions based on the writing style of questions from the Spider dataset. We also built queries for the natural language question dataset, but we used the same queries we built for claims for the mimicked questions set.

We then used SmBop to get the predicted queries for all three datasets. Experiment results show that the performance of the text-to-SQL parser is directly related to the writing style of the text. While SmBop obtained an EM of 70% for the mimicked dataset, it got a 0 EM score for the claims and natural language question datasets.

4.2 Related Works

The interdisciplinary nature of misinformation has raised a considerable response from the academic research communities, including computer science, political science, and journalism. In this section, we review related work in two areas: 1) the automatic verification of factual claims (aka fact-checking); and 2) the text-to-SQL models.

4.2.1 Fact-checking

In recent years, the growing interest in fact-checking has resulted in the development of various computational methods and tools in vetting claims by using supporting, refuting, and related evidence sentences from web documents [16, 17, 18, 19], making use of knowledge bases by associating claim entities with KB properties [20, 21, 22, 23, 24, 25, 26], and translating claims into verification queries to run on relational databases [27, 28, 29].

The development of claim checking models using relational databases has received much less attention compared to the aforementioned approaches. The lack of claim - SQL query pairs' dataset and the difficulty in automatically translating claims to SQL queries may have played a role in this.

Jo et al. [27, 28] introduced the AggChecker to translate numerical claims to SQL queries. Given a claim and its associated data set, AggChecker constructs candidate SQL queries based on calculated relevance scores between pre-defined SQL query fragments and the claim keywords. AggChecker then uses an expectation-maximization algorithm to

compute the probabilities of query candidates. Finally, the AggChecker decides if a claim is likely to be wrong based on its most likely query.

Karagiannis et al. [29] introduced Scrutinizer, a system that uses four classifiers to extract the fragments of the final SQL query from a given claim. The three classifiers work to identify essential elements of each query, such as primary keys values, names of attributes, and relevant relations. The final classifier identifies a generic formula with variables in the place of keys and attribute values. If Scrutinizer cannot predict an element with high confidence, it requires users to build the query.

4.2.2 Text-to-SQL

The text-to-SQL problem has been studied for decades in database and NLP fields. However, the recent advances in deep learning networks [74] and pretrained language models [75, 87] have given rise to a renewed interest in this area. Additionally, with the creation of two new large-scale datasets, such as WikiSQL [78] and SPIDER [79], and new language models pretrained specifically for handling structured data, such as TaBERT [76] and GraPPa [77], several new models [80, 81, 82, 83, 84, 85, 86] have been introduced recently.

To the best of our knowledge, our study is the first to investigate the efficacy of using a text-to-SQL model in claim translation. Jo et al. [27, 28] used NaLIR [88], a natural language database query interface, as a baseline model for performance comparison. To use NaLIR, they employed a question generation tool to translate claims into questions and conclude that NaLIR translated less than 5% of claims while throwing exceptions for the rest of the claims during the translation process.

4.3 Challenges

Deploying a text-to-SQL based model to automate verification of factual claims comes with many challenges: those that are unique to the fact-checking process, and those that are common across all text-to-SQL models.

4.3.1 Lack of authoritative data:

While it is challenging to obtain reliable and authoritative data that helps debunk misinformation, only a small portion of such data is in the form of structured data that can be easily leveraged in fact-checking tools. Thus, the lack of availability of structured data limits the effectiveness of automated fact-checking.

4.3.2 Complete data:

In fact-checking, the data available is critical when a claim is made and needs to be verified. Therefore, for claim verification, complete authoritative data is a necessary requirement.

4.3.3 Translating text to query:

Translating a textual claim to a structured query is yet another challenge due to the complex nature of human language. Based on our observations from already fact-checked claims, claim statements often contain multiple claims that can be verified on the same database or on different databases. The challenge here is to identify which parts of the claim need to be utilized in query generation. For instance, consider the following two voting-related claims: “Congressman DeSantis voted to cut Social Security and Medicare and voted to increase the retirement age.” and “Thirteen Democratic senators voted against cheaper medicines and took millions from big pharma since 2011.” While both of the aforementioned statements consist of two claims each, only the first statement can be fully

verified on a congressional voting database since both of the claims are voting-related. In contrast, only one of the claims of the second statement can be verified on the congressional voting database. Hence, a fact-checking system must identify and verify each claim for a multi-claim statement to assign a verdict to the entire statement.

Since natural language is intrinsically ambiguous, a statement can be interpreted in various ways. Consider the COVID-19 related claim, "The death rate in the US is higher than in Germany." A study [2] from Karagiannis et al. [29] has shown that it can be interpreted in four different ways: based on total death vs. population for today; based on total death vs. total confirmed cases for today; based on total death vs. population for the entire month of July 2021; based on total death vs. total confirmed cases for the entire month of July 2021. Each interpretation associates the statement with a different query and can potentially result in a different verdict in each case.

4.3.4 Limited metadata:

The metadata of a database is constrained to table names and attribute names, resulting in a limited vocabulary for expressing a factual claim. Human language is rich in terms of vocabulary and expressivity and hence translating a statement (i.e., factual claim) to a corresponding SQL query is challenging. Therefore, a text-to-SQL model is required to make inferences and convert a claim to a corresponding accurate query. This has its own set of challenges, such as linking table names and attribute names with the properties of the claims.

4.4 Experimental Setup

In this study, we conducted a set of experiments to assess the efficacy of a current state-of-the-art text-to-SQL model on generating SQL queries for claim verification. We

[2] https://coronacheck.eurecom.fr/en

employ the current state-of-the-art text-to-SQL model, [3] SmBoP, a semi-autoregressive bottom-up semantic parser [86], for SQL query generation. We generated three small-scale datasets to evaluate the performance of SmBop. We describe the dataset generation process in the following section.

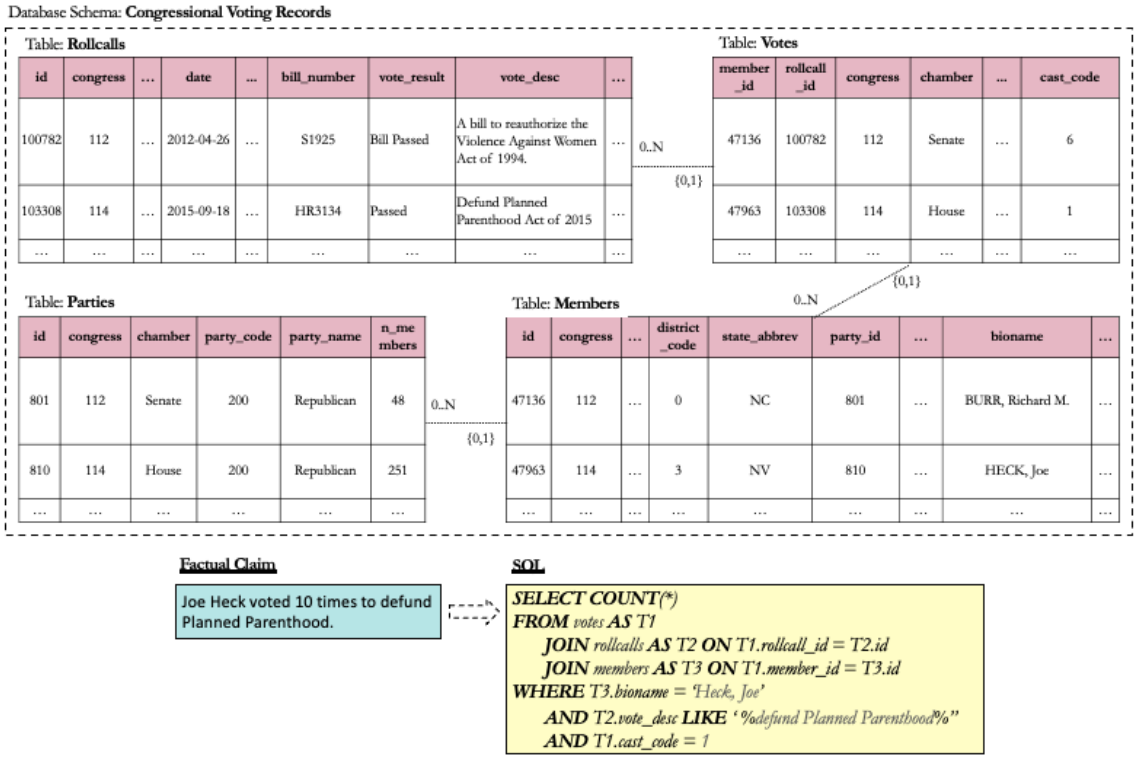


Figure 4.2: The inputs include a database schema and a factual claim. The output is a predicted SQL query.

[3] As of July 2nd, 2021, SmBoP has the best performance on the SPIDER leaderboard: https://yale-lily.github.io/spider

4.4.1 Dataset Construction

Congressional voting database: We created a database of congressional voting records from csv files. [4] The Figure 4.2 shows the database schema of the congressional voting records. The database consists of the following four tables.

- **Rollcalls:** This table contains data about each rollcall taken by Congress in the selected chamber such as "House", "Senate", or "President".
- **Members:** Members table has biographical information about members of congress.
- **Votes:** This table contains basic information about how each member in the selected chamber(s) and congress(es) voted on each vote.
- **Parties:** This final table contains biographical information for congressional parties in the selected chamber(s) and congress(es).

We created three sets of text and SQL query pairs.

1. Claim to SQL (c-sql) set: We analyzed a set of 104 voting related fact-checked claims from PolitiFact [5], a fact-checking organization. We then removed claims that are not related to congressional voting. We used the remaining 65 claims and manually built SQL queries for each of them. Each of those SQL queries were built in a way that the query results can be used to verify the claims. The Figure 4.3 shows a sample claim and its corresponding SQL query.

2. Natural language question to SQL (nq-sql) set: We translated each claim into a natural language question using a t5-small [89] model [6] trained for end-to-end question generation task. For a given input text, the model will generate multiple questions. We used factual claims as input and among the output questions we selected those that were about voting. Then, for each question, we manually built its corresponding SQL query. Figure 4.4 presents

[4] https://voteview.com/data

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

[6] https://huggingface.co/valhalla/t5-small-e2e-qg

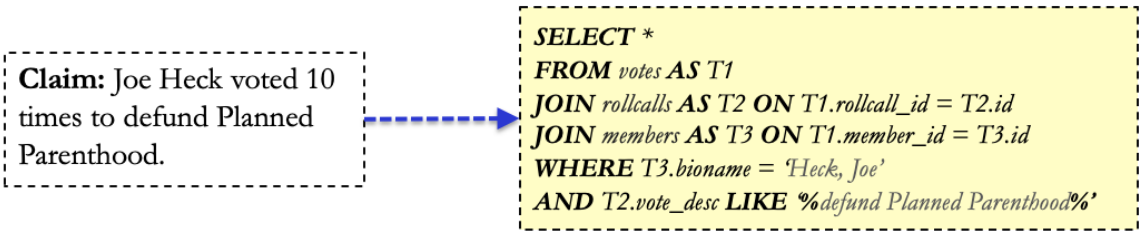


Figure 4.3: A factual claim and its corresponding SQL query

a sample question, its corresponding natural language question, and the generated SQL query for the question.

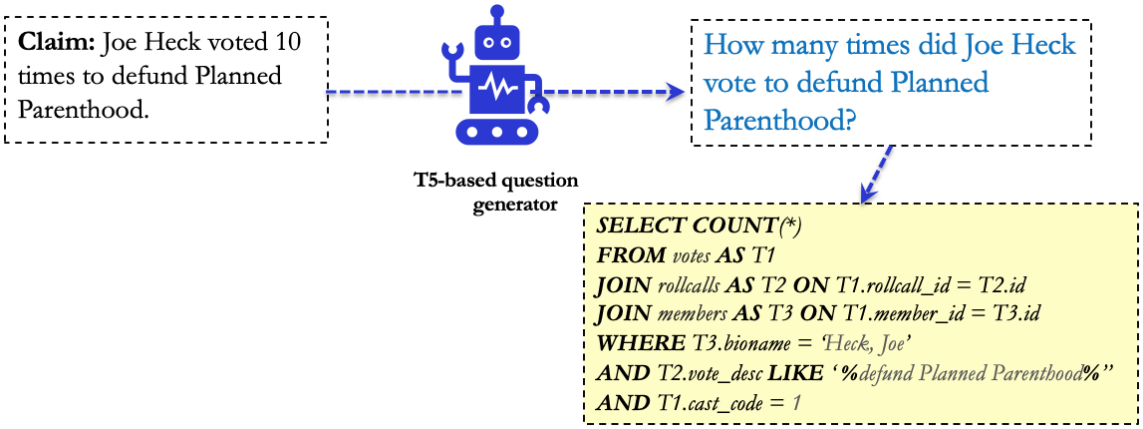


Figure 4.4: A factual claim is automatically translated into a natural language question. A SQL query is built for the question.

3. Mimicked question to SQL (mq-sql) set: We first analyzed a subset of questions from the Spider dataset. We then generated a new dataset by replacing each factual claim of the c-sql dataset with a natural language utterance written by mimicking the writing style of the utterance from the Spider dataset. While we substituted claims with natural language utterances in this new dataset, we kept the SQL queries from the c-sql dataset.

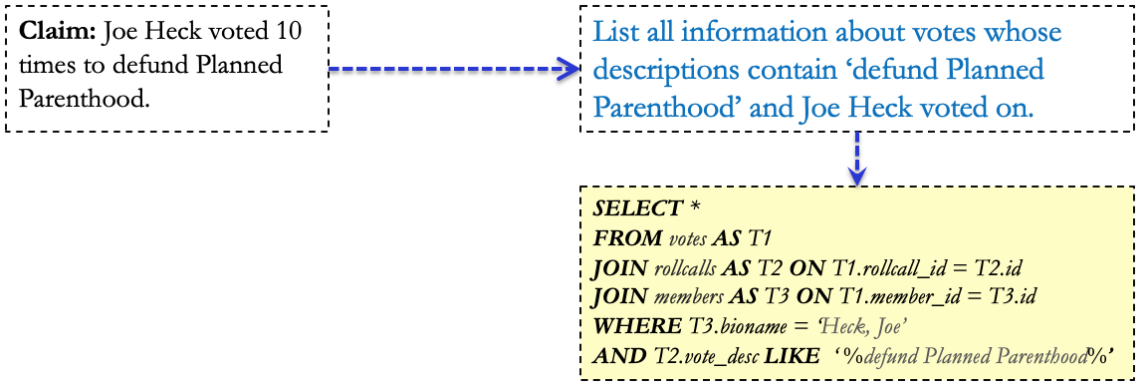


Figure 4.5: A factual claim is manually translated into a mimicked question. A SQL query is built for the question.

4.4.2 Evaluation Metrics

Exact Matching: This metric evaluates the structural correctness of the predicted SQL by comparing each SQL component in the predicted query with regard to the gold query. The predicted query is accepted as correct only if all of the components are correct. This metric disregards the predicted values.

Execution Accuracy: This metric runs the predicted query on its corresponding executable SQLite database and checks if the execution results of the predicted SQL match the results of the gold query.

4.4.3 Results

Table 4.1 shows test results of SmBop on c-sql, nq-sql, mq-sql datasets. SmBop obtains an EM of 70% for the mq-sql dataset, only 0.5% higher than its reported performance on the SPIDER test set. However, it gets a 0 EM score for the c-sql and nq-sql datasets. This result shows that the performance of the text-to-SQL parser is directly related to the writing style of the dataset. SmBoP gets an execution accuracy of 0 for all three datasets. We find that while the "bioname" attribute of the "members" table saves the value in the format of

"last name, first name" (e.g., "Heck, Joe"), the predicted value's format is " first name last name" (e.g., "Joe Heck"). Due to the difference in the format, the predicted queries generate no results.

Table 4.1: Results on Congressional Voting dataset in terms of Exact Matching (EM) and Execution Accuracy (EA).

	c-sql	**nq-sql**	**mq-sql**
EM	0.0	0.0	**0.70**
EA	0.0	0.0	0.0

Table 4.2 shows partial matching results of SmBoP on the three datasets for SELECT, WHERE, and KEYWORDS components. The keywords component contains all SQL keywords without column names and operators. Results show that SmBop fails at correctly predicting any of the components for the c-sql dataset. However, it obtains a partial matching of 79% and 100% for the SELECT component for c-sql and mq-sql datasets, respectively.

Table 4.2: Partial matching results on Congressional Voting dataset in terms of Precision, Recall, F1.

	Precision			**Recall**			**F1**		
	c-sql	**nq-sql**	**mq-sql**	**c-sql**	**nq-sql**	**mq-sql**	**c-sql**	**nq-sql**	**mq-sql**
select	0.0	0.84	1.0	0.0	0.75	1.0	0.0	0.79	1.0
where	0.0	0.0	0.72	0.0	0.0	0.72	0.0	0.0	0.72
keywords	0.0	0.0	0.88	0.0	0.0	0.88	0.0	0.0	0.88

CHAPTER 5

OTHER STUDIES

In this chapter, we present two studies. We first introduce ClaimPortal, our web-based platform for monitoring, searching, checking, and analyzing English 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 and a claim spotter. The claim matcher finds known fact-checks matching any given tweet. The claim spotter 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 search interface that assists its users to sift through these factual claims in tweets using filtering conditions on dates, Twitter accounts, content, hashtags, check-worthiness scores, and types of claims. We also explain in detail our study of categorizing tweets by the type of factual claims they promote using semantic frames.

We then present a dataset of claims from all U.S. presidential debates (1960 to 2016) along with human-annotated check-worthiness label. We argue that the research community lacks a large labeled dataset of claims to leverage in claim detection tasks. To address this need, we provide a large dataset of $23,533$ sentences where each sentence is categorized into one of three categories; non-factual statement, unimportant factual statement, and check-worthy factual statement. We explain our data collection process in detail.

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

5.1.1 Introduction

The problem of unchecked claims is exacerbated on social media. On the one hand, it is unlikely fact-checkers are able to check every social media post, due to limited resources and the sheer volume of data. [1] 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. [2]

To fulfill this gap, we built ClaimPortal, a web-based platform with a user-friendly interface that helps monitor, search and check English factual claims on Twitter. It is integrated with fact-checking tools, including a claim matcher and a claim spotter. The claim matcher finds known fact-checks matching any given tweet. The claim spotter scores each claim and the corresponding tweet based on their check-worthiness, i.e., how important it is to fact-check them. ClaimPortal boosts its usability with various filtering conditions

---

* This section is adapted from [66]: Majithia, S., **Arslan, F.**, Lubal, S., Jimenez, D., Arora, P., Caraballo, J., and Li, C. (2019). ClaimPortal: Integrated Monitoring, Searching, Checking, and Analytics of Factual Claims on Twitter. *In Proceedings of ACL 2019.*

1 https://mashable.com/article/snopes-stops-fact-checking-for-facebook/

2 http://politicalbots.org/?p=1064

on dates, Twitter accounts, content, hashtags, check-worthiness scores, and types of claims. ClaimPortal also categorizes tweets by the type of claim they promote. ClaimPortal is available at https://idir.uta.edu/claimportal.

5.1.2 System Architecture and Components

5.1.2.1 System Architecture

ClaimPortal is composed of a front-end web based GUI, a MySQL database, an Elasticsearch [3] search engine, an API, and several decoupled batch data processing components (Figure 5.1). The system operates on two layers. The *front-end presentation layer* 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 data collection and computation layer* performs pre-processing of tweets, computing check-worthiness scores of tweets using the public ClaimBuster API [12], Elasticsearch batch insertion, detecting claim types of tweets, and finding similar fact-checked claims for each tweet, using ClaimBuster API. ClaimPortal stays up-to-date with current tweets by periodically calling the Twitter REST API.

5.1.2.2 Monitoring, Processing, and Storing Tweets

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

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

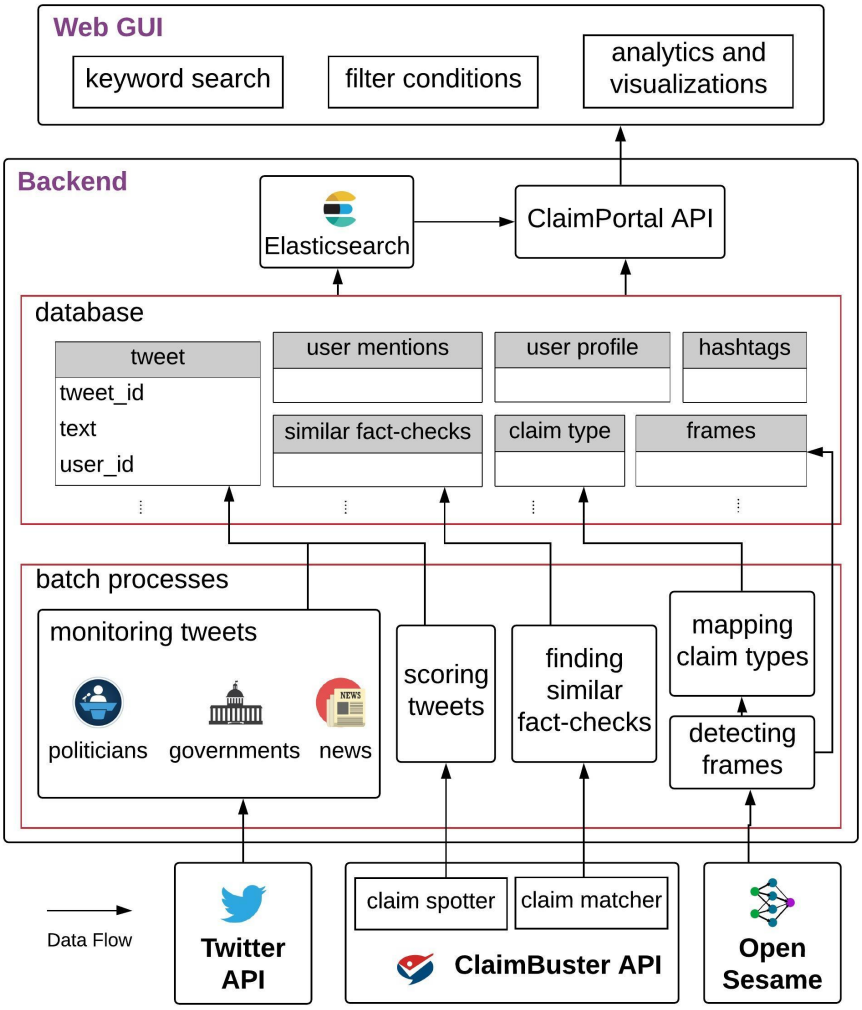


Figure 5.1: *ClaimPortal* system architecture.

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 working on substantially expanding the curated list of Twitter handles.

Table 5.1: Claim types and their corresponding FrameNet frames. Frames written in italics are the ones that we have introduced in Chapter 3.

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*, *Taking Side Consistency*
Speech	*Statement*, Affirm or deny, Telling

ClaimPortal's back-end 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 small and yet sufficient enough to provide all necessary information for the portal. This is achieved through the ClaimPortal API. The API is a web service designed using Python and the Flask [4] micro-framework. It provides end points for loading tweets on the GUI, search for hashtags, and search for users in applying from-user and user-mention filters. Based on

[4] http://flask.pocoo.org

the keyword search and filters requested by a user, the API queries the database to find the resulting list of tweet IDs and returns the list as a JSON response. A tweet ID is a unique number assigned to a tweet by Twitter. By using Twitter's card API [5] 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, hashtags 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 updating the Elasticsearch index.

5.1.2.3 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, [6] a well-known fact-checking tool, developed by our research group, that is being used by professional fact-checkers on a regular basis [90]. ClaimBuster [12, 9] is a classification and ranking model trained on a large human-labeled dataset of statements from past U.S. presidential debates.

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.

[5] https://developer.twitter.com/en/docs/tweets/optimize-with-cards

[6] https://idir.uta.edu/claimbuster/

5.1.2.4 Claim Type Detection

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 [34] and created several new frames specifically for factual claims. We then adopted the study of mapping frames to event types [91].

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 [7] and 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 [62]. The corpus of the newly-defined frames along with their annotated exemplary sentences is publicly available. [8] We used open-sesame [92], 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.

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

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

Claim type mapper: In [91], 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 5.1.2.4, we extended this list by introducing four new event types, namely *Comparison, Quantity, Stance*, and *Speech*, and their corresponding frames (Table 5.1). 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 entities 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.

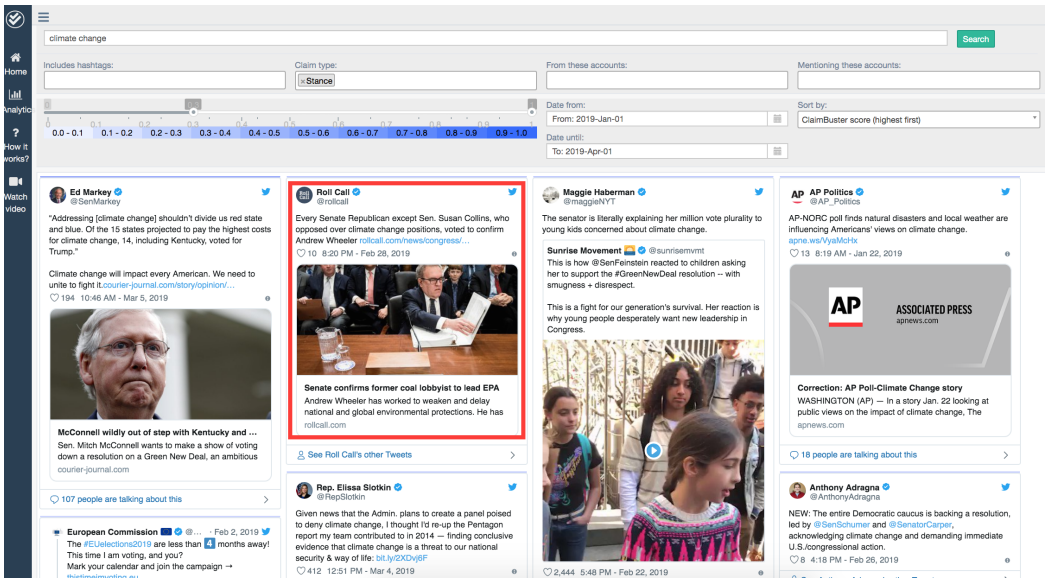
5.1.2.5 Claim Matcher

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. [9]

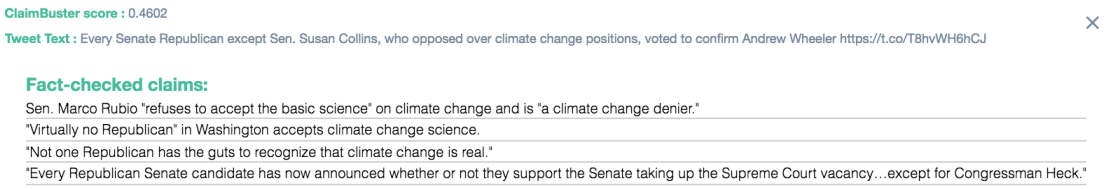
ClaimPortal leverages the claim matching function in the ClaimBuster API. The fact-check repository is composed of the Share-the-facts [10] fact checks as well as fact checks collected from several fact-checking organizations like PolitiFact, Snopes, factcheck.org, Washington Post, etc. The system measures 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.

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

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



(a)



(b)

Figure 5.2: (a) *ClaimPortal* user interface. (b) Similar fact-checks for the highlighted tweet in Figure (a).

5.1.3 User Interface Features

ClaimPortal enables a user to sift through the tweets using multiple filters. The important filters are as follows. **(1) Keyword search:** It allows user to make a text-based search by typing the desired keywords such as "climate change" in the search input area at the top. This displays all the tweets pertaining to the search criteria, "climate change". **(2) Hashtags:** It allows users to further filter tweets by hashtags such as "#116thCongress" or "#2020". **(3) Claim type:** It enables users to search for tweets with a specific claim type, e.g., *Conflict* or *Stance*. **(4) From:** It looks for tweets posted by a particular user handle, e.g., "@realDonaldTrump". **(5) Mentions:** The search results can be filtered further by user

mentions (i.e., using "@" to tag a user in a tweet, e.g., "@POTUS"). **(6) ClaimBuster score:** ClaimPortal also offers a slider to filter results based on a ClaimBuster score range. The result tweets are automatically updated as the slider is moved. **(7) Date range:** Additionally, the portal offers a date picker to filter tweets based on their creation dates.

Figure 5.2a shows ClaimPortal user interface with the 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 from 0.3 to 1.0, and a date range from January 1, 2019 to April 1, 2019. Moreover, the ClaimPortal shares previously fact-checked claims with users by displaying matching fact-checks after a tweet's card view is clicked at. Figure 5.2b depicts the matching fact-checks of the highlighted tweet in Figure 5.2a.

5.2 A Benchmark Dataset of Check-Worthy Factual Claims ¶¶

5.2.1 Introduction

One of the key elements in the fact-checking process is automatically assessing the check-worthiness of a a piece of information. Such an assessment can not only assist the journalists with providing them with the most check-worthy claims from an interview or debate but also lessens the potential of human bias in claim selection. However, to have an accurate automated check-worthiness assessment, it is imperative to have a carefully annotated ground-truth dataset that can fuel a machine learning algorithm to predict the check-worthiness of a statement.

In this section, we present a dataset of claims from all U.S. presidential debates (1960 to 2016) along with human-annotated check-worthiness label. It contains $23,533$ sentences where each sentence is categorized into one of the three categories- non-factual statement,

¶¶ This section is adapted from [93]: **Arslan, F.**, Hassan, N., Li, C., and Tremayne, M. (2020). A benchmark dataset of check-worthy factual claims. *In Proceedings of ICWSM*.

unimportant factual statement, and check-worthy factual statement. These sentences have been labeled by 101 coders over a 26 months period in multiple phases.

This dataset has been used to develop the first-ever end-to-end automated fact-checking system, ClaimBuster [12, 94]. It has been used to study how an automated check-worthiness detector fares compared to human judgements [95]. Also, it has been used to deliver check-worthy factual claims filtered from a variety sources including PolitiFact, [1] one of the leading fact-checking organization in the United States [90]. Through this paper, we make the dataset publicly available.

In the following sections, we describe the preparation process of the dataset, present descriptive statistics of the dataset, suggest possible use cases, and explain different fairness policies we have followed while developing this dataset.

5.2.2 Related Works

Researchers have attempted to prepare datasets of check-worthy factual claims to assist automated fact-checking. For instance, Nakov et al. [37] developed a dataset of check-worthy factual claims from the 2016 U.S. presidential debate. To determine the check-worthiness of statements, the authors used available fact-checks of the debate by a fact-checking organization, FactCheck.org. If FactCheck.org has checked a statement from the debate, the dataset labels that statement as check worthy; otherwise not. While this strategy ensures that their check-worthy statements are indeed picked by professional fact-checkers it does not resolve the question of whether selection bias of a single organization may have tainted the quality of the dataset. Our strategy for annotation considers input from multiple high-quality, trained coders. This decreases the chance of having a dataset with a bias towards certain ideology. Also, unlike the dataset of [37], that had 2016 debates and

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

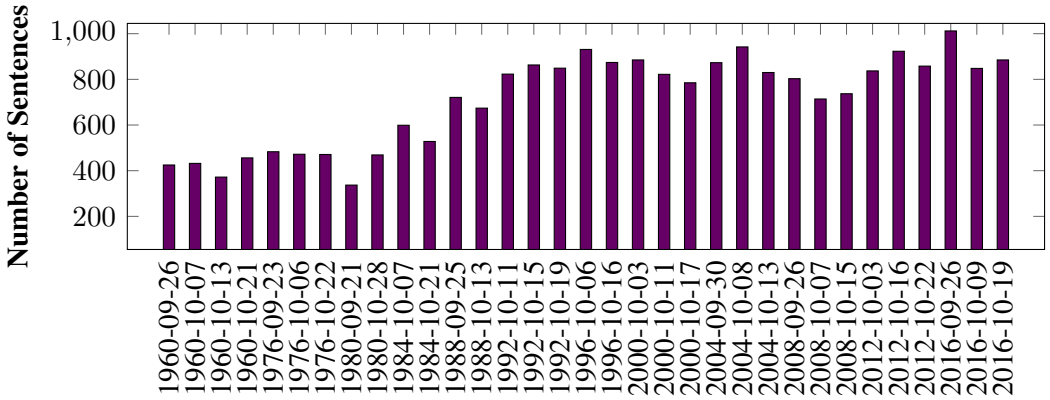


Figure 5.3: Sentence distribution among presidential debates

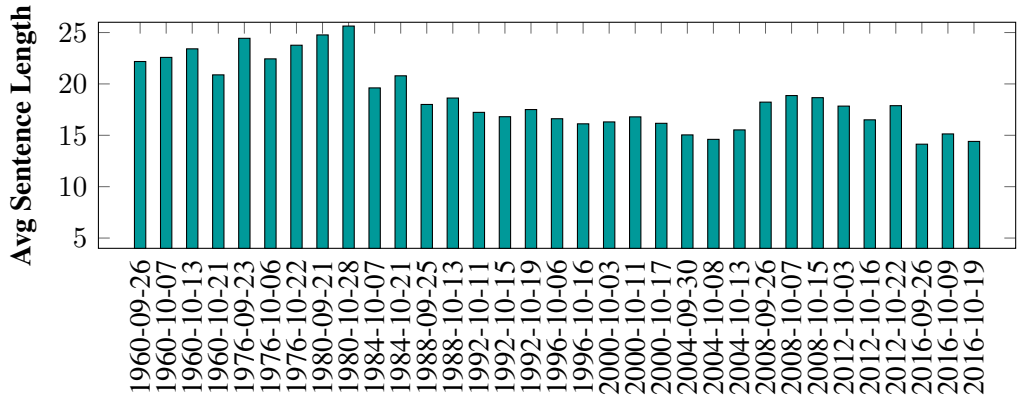


Figure 5.4: Average sentence length in words per debate

several political speeches of that time, we annotated all the U.S. general election presidential debates since 1960.

Patwari et al. [7] prepared another dataset of check-worthy factual claims by combining the fact-checks of 15 2016 U.S. election primary debates from 9 fact-checking organizations (e.g., Fox News, NPR, CNN). Although having inputs from a range of fact-checking organizations reduces the chance of having a biased sample the dataset becomes specific to certain issues that were relevant during the 2016 presidential election. As our dataset covers a longer time-period, over 50 years, it captures more general issues and patterns that are relevant for assessing the check-worthiness of a broader array of claims.

5.2.3 Transcript Extraction and Processing

Candidate sentences were extracted from U.S. presidential debate transcripts. [2] The first general election presidential debate was held in 1960. Since then, there were a total of 15 presidential elections from 1960 to 2016. In 1964, 1968, and 1972, no presidential debate was held. There were 2 to 4 debate episodes in each of the remaining 12 elections. A total of 33 debate episodes spanned from 1960 to 2016. There are $32,072$ sentences spoken in these debates. We applied the following steps to prepare the candidate sentences to be labeled.

1. Using parsing rules and human annotation, the speaker of the each sentence was identified. $26,322$ sentences are spoken by the presidential candidates, $4,292$ by the debate moderators, and $1,319$ by the questioners. There are 139 sentences without a speaker name which were voice-over announcers at the start of the debate (i.e., "September 26, 2008.", "The First McCain-Obama Presidential Debate").
2. We only focused on the sentences spoken by the presidential candidates. Therefore, sentences spoken by the debate moderators, the questioners, and the announcers were discarded from further labeling.
3. Another processing step was performed to filter very short sentences. We removed sentences shorter than 5 words. In total, $2,789$ sentences were discarded, which represent 8.69% of the original dataset.

The resulting dataset (henceforth referred to as the *ClaimBuster* dataset) contains $23,533$ labeled sentences. Figure 5.3 shows the distribution of the sentences among 33 debate episodes and Figure 5.4 depicts the average length of sentences per debate. These figures show that although the number of spoken sentences increased in recent debates, they got shorter comparing to earlier debates.

[2] https://www.debates.org/voter-education/debate-transcripts/

5.2.4 Annotation Procedure

5.2.4.1 Annotation Guideline

We categorize the sentences from the *ClaimBuster* dataset into three groups. Below, we define each category, along with examples.

Check-worthy Factual Sentence (CFS): These sentences contain factual claims that the general public will be interested in learning about their veracity. Journalists look for these types of claims for fact-checking. Some examples are:

- In the last month, we've had a net loss of one hundred and sixty-three thousand jobs.
- We've spent $4.7 billion a year in the State of Texas for uninsured people.

Unimportant Factual Sentence (UFS): These are factual claims but not check-worthy. In other words, the general public will not be interested in knowing whether these sentences are true or false. Fact-checkers do not find these sentences as significant for checking. A few examples are as follows:

- I am a son of a Methodist minister.
- Just yesterday, I was in Toledo shaking some hands in a line.

Non-factual Sentence (NFS): These sentences do not contain any factual claims. Subjective sentences (opinions, beliefs, declarations) and many questions fall under this category. Below are some examples.

- The worst thing we could do in this economic climate is to raise people's taxes.
- I think the Head Start program is a great program.

5.2.4.2 Data Collection Platform

We used our in-house data collection website [3] to collect the ground-truth labels of the sentences. Figure 5.5 shows its interface. A participant is presented one sentence at a

[3] http://idir.uta.edu/classifyfact_survey

time, and it is randomly selected from the set of sentences not seen by the participant before. The participant can assign one of three possible labels [NFS, UFS, CFS] for the sentence.

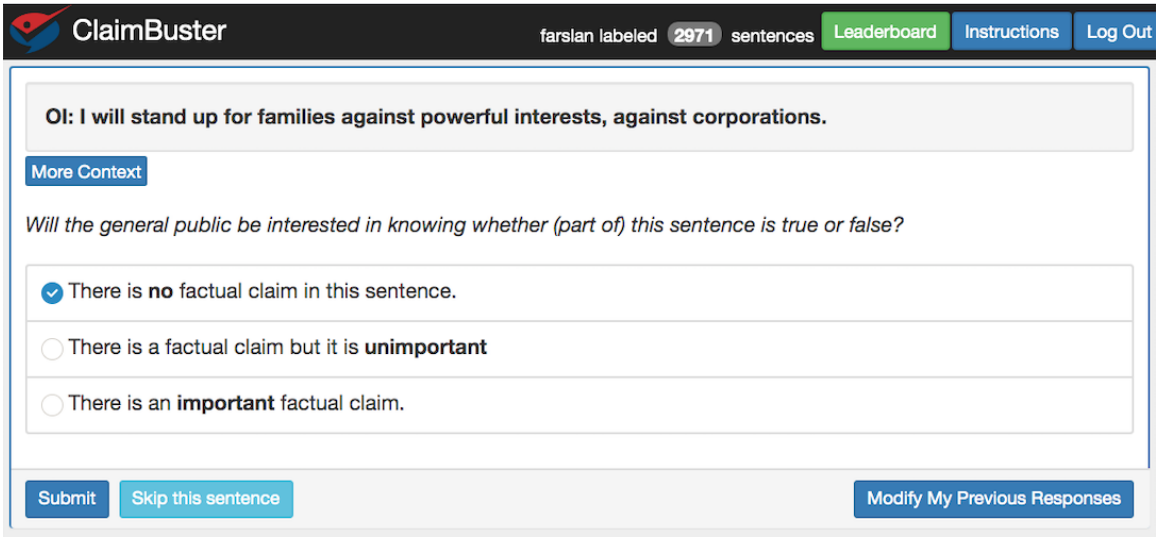


Figure 5.5: Data collection interface

5.2.4.3 Quality Control

We selected 1032 sentences from all the sentences to create a ground-truth dataset. Three experts agreed upon the labels of these sentences: 731 NFS, 63 UFS, 238 CFS. We used this ground-truth dataset to detect spammers and low-quality participants for ensuring high-quality labels. On average, one out of every ten sentences given to a participant (without letting the participant know) was randomly chosen to be a screening sentence. First, a random number decides the type (NFS, UFS, CFS) of the sentence. Then, the screening sentence is randomly picked from the pool of screening sentences of that particular type. The degree of agreement on screening sentences between a participant and the three experts is one of the factors in measuring the quality of the participant. For a screening sentence, when a participant's label matches the experts' label, s/he is rewarded with some points. If it does not match, s/he is penalized. We observe that not all kinds of mislabeling has

equal significance. For example, labeling an NFS sentence as a CFS is a more critical mistake than labeling a UFS as a CFS. We defined weights for different types of mistakes and incorporated them into the quality measure.

Formally, given $SS(p)$ as the set of screening sentences labeled by a participant p, the labeling quality of p (LQ_p) is

$$LQ_p = \frac{\sum_{s \in SS(p)} \gamma^{lt}}{|SS(p)|}$$

where γ^{lt} is the weight factor when p labeled the screening sentence s as l and the experts labeled it as t. Both $l, t \in \{NFS, UFS, CFS\}$. We set $\gamma^{lt} = -0.2$ where $l = t$, $\gamma^{lt} = 2.5$ where $(l, t) \in \{(NFS, CFS), (CFS, NFS)\}$ and $\gamma^{lt} = 0.7$ for all other combinations. The weights are set empirically. If $LQ_p \leq 0$ for a participant p and p labeled at least 50 sentences, we designate p as a top-quality participant. A total of 405 participants contributed in the data collection process so far. Among them, 101 are top-quality participants. Throughout data collection process, the top-quality participants encountered screening sentences 9986 times; 5222 NFS, 1664 UFS, and 3100 CFS. They chose incorrect labels 511 (5%) times.

5.2.5 Dataset Description

5.2.5.1 Dataset Statistics

We collected $88,313$ labels among which $62,404$ (70.6%) are from top-quality participants. There are $22,281$ (99.02%) sentences which satisfy the above stopping condition. Table 5.2 shows the distribution of the classes in these sentences. The remaining 220 sentences, though, received many responses from top-quality participants, the labeling agreement did not satisfy the stoping condition. We assign each sentence the label with the majority count. Figure 5.6 depicts the class distribution of sentences among 33 presidential debates, including all $22,501$ human-annotated sentences and $1,032$ expert labeled screening sentences.

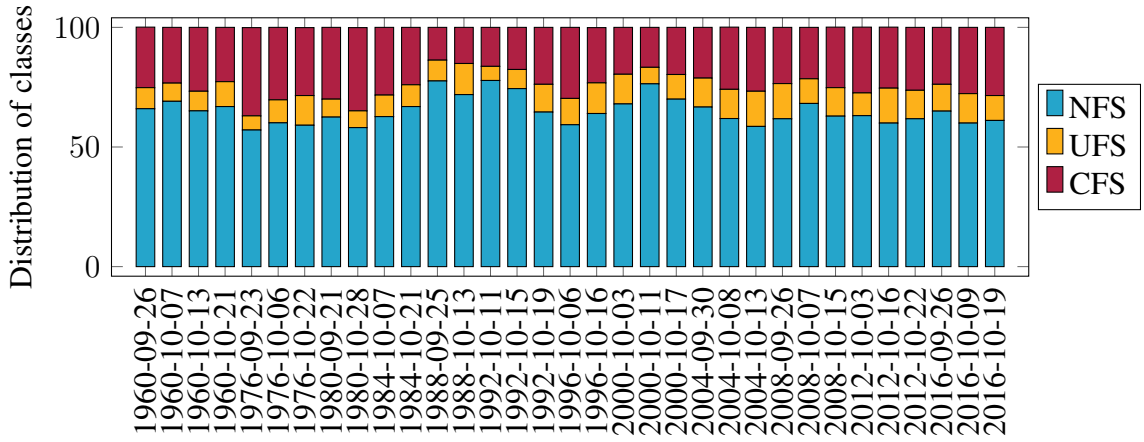


Figure 5.6: Class distribution per debate

Table 5.2: Distribution of sentences over classes

Assigned label	#sent	%
CFS	5,318	23,87
UFS	2,328	10.45
NFS	14,635	65.68
total	22,281	100.00

During the data collection process, we advised the participants to skip the sentences that they are not confident in assigning a label. We analyzed the correlation between the number of sentences and the number of times they were skipped by the top-quality participants. We found that 17,874 (79.4%) sentences were not skipped by any of the top participants, while the remaining 4,627 (20.6%) sentences were skipped at least once. This observation indicates that participants found one in every five sentences challenging. Table 5.3 presents the distribution of these 4,627 sentences based on the frequency of them being skipped. For instance, 742 sentences were skipped by any of the two top participants. One interesting observation is that the length of the sentences increased proportionally with the increasing number of skips.

We further analyzed each claim type according to the number of responses we obtained from the top-quality participants to assign each sentence a label. Table 5.4 depicts the

Table 5.3: Sentence distribution in terms of frequency of user skip

#skip	1	2	3	4	5	6	7
#sentence	3686	742	155	32	7	3	2
#words(avg)	19.3	21.3	25.2	25	26.7	35	62.5

distribution of responses over sentences — the frequency of responses spans from 2 to 18. The vast majority of the sentences (93%) were labeled by 2 or 3 participants, meaning that at least two of the participants agreed upon the label. Four or five participants labeled the 4.3% of the remaining 7% of the sentences, indicating that at least three participants gave the same response. However, the participants were challenged to agree on the label of 620 (2.7%) sentences as the number of the responses varies from 6 to 18.

5.2.6 Possible Use Cases

The claim detection task is to detect claims worthy of fact-checking from natural language statements, and can be approached in two ways. One of the approaches is to identify if a sentence comprises a factual claim aside from its check-worthiness. The second approach takes the check-worthiness of the claim into consideration. In the following sections, we argue how these two claim detection approaches can utilize the *ClaimBuster* dataset.

Table 5.4: Frequency distribution of participants' responses over each class type

#responses	**#sentences**	**NFS**	**UFS**	**CFS**
2	13057 (58%)	9388 (63.9%)	845 (35.1%)	2824 (52.2%)
3	7865 (35%)	4545 (30.9%)	1192 (49.6%)	2128 (39.3%)
4	329 (1.5%)	224 (1.5%)	40 (1.7%)	65 (1.2%)
5	630 (2.8%)	309 (2.1%)	152 (6.3%)	169 (3.1%)
6-10	295 (1.3%)	125 (0.9%)	70 (2.9%)	100 (1.8%)
11-18	325 (1.4%)	94 (0.6%)	104 (4.3%)	127 (2.3%)
Total	22501	14685	2403	5413

Factual Claim Detection: This approach formulates the task as a binary classification task that identifies a sentence as either containing a factual claim (FC) or not containing a factual claim (NFC). This task can make use of the ClaimBuster dataset by combining UFS sentences and CFS sentences into FC sentences and using NFS sentences as NFC sentences. Then, a binary classifier can be trained on the FC and NFC sentences and applied to future sentences.

Check-worthy Claim Detection: In order to prioritize the most check-worthy claims over less check-worthy ones, a check-worthiness score, which is the probability that a sentence belongs to the CFS class, is required. To this aim, this approach models the claim detection problem as a classification and ranking task. Given a sentence, a machine learning model or neural network model trained on the ClaimBuster dataset calculates a check-worthiness score that reflects the degree by which the sentence belongs to CFS.

5.2.7 FAIRness

In this section, we explain how we have made the ClaimBuster dataset adhere to the "FAIR" Facets: Findable, Accessible, Interoperable, and Re-usable. To be *Findable* and *Accessible*, we make the dataset publicly available through Zenodo, [4] a dataset sharing platform, allowing the complete dataset to be downloaded. The dataset files are provided in CSV (Comma Separated Values) format that can be utilized by any applications and exported to other data formats. The dataset is supplemented with a readme file explaining each data file in detail to optimize the re-use of the dataset.

[4] https://zenodo.org/

CHAPTER 6

CONCLUSIONS

The proliferation of misinformation has elicited a number of responses from researchers from various disciplines to assist fact-checkers by creating several automated fact-checking tools, datasets, services, and applications. In this dissertation, we focus on studying factual claims and make the following contributions to assist the automated fact-checking efforts:

- Understanding a factual claim and parsing the content of the claim to extract its attributes are challenging. We propose a way to represent claims in a structured form to capture various aspects of claims, such as entities involved, their relationships, quantities, points and intervals in time, comparisons, and aggregate structures. We use semantic frames for the representation of factual claims. We create a set of 11 new semantic frames and adopt nine FrameNet frames. We further create a dataset of frame-annotated claims and a publicly available annotation tool (Chapter 3).
- To verify a factual claim over a relational database, it is necessary to translate it into a SQL query. However, automatically translating claims to SQL queries is hard. We conduct a preliminary investigative study: (a) to reveal challenges in claim translations and (b) to assess the efficacy of applying a state-of-the-art text-to-SQL parser in translation. Our experiment results show that the performance of the text-to-SQL parser is directly related to the writing style of the dataset (Chapter 4).
- The problem of unchecked claims is exacerbated on social media. We build ClaimPortal, a web-based platform. The ClaimPortal enables users to monitor, search, and check English factual claims on Twitter. ClaimPortal provides an intuitive and convenient

search interface that assists users in sifting through factual claims in tweets via filtering conditions on dates, Twitter accounts, content, hashtags, check-worthiness scores, and types of claims. To identify claim types, we propose a semantic-frame-based model (Chapter 5.1).

- One of the critical elements in the fact-checking process is automatically assessing the check-worthiness of a piece of information. Such an assessment can assist the journalists by providing them with the most check-worthy claims from an interview or debate and reduces human bias from seeping into claim selection. However, to have an accurate automated check-worthiness assessment, it is imperative to have a carefully annotated ground-truth dataset that can fuel a machine-learning algorithm to predict the check-worthiness of a statement. We create a dataset of claims from all U.S. presidential debates (1960 to 2016) along with the human-annotated check-worthiness label. It contains $23,533$ sentences where each sentence is categorized into one of the three categories- non-factual statements, unimportant factual statements, and check-worthy factual statements. This dataset can be leveraged to build computational methods to identify claims worth fact-checking from the myriad sources of digital or traditional media (Chapter 5.2).

APPENDIX A

Corpus of Factual-claim Specific Frames and Frequencies of Frame Instances

Table A.1: Frequencies of frame instances per lexical unit (LU) in the corpus of factual-claim specific frames (*continued on next page*)

STATEMENT	acknowledge.v (3), acknowledgment.n, add.v, address.v, admission.n, allegation.n (6), allege.v (4), allow.v, announce.v (19), announcement.n (4), assert.v (4), assertion.n, attest.v (1), aver.v, avow.v, avowal.n, be like.v, caution.n, caution.v, challenge.v, claim.n (18), claim.v (11), comment.n (5), comment.v (3), concession.n, confirm.v (3), conjecture.n, conjecture.v, contend.v, contention.n, declaration.n, declare.v (5), denial.n, describe.v (7), detail.v, exclaim.v, exclamation.n, explain.v (2), gloat.v, explanation.n, hazard.v, insist.v, insistence.n (1), maintain.v, mention.n, mention.v (9), message.n, note.v (2), observe.v (1), pout.v, preach.v, proclaim.v, proclamation.n, profess.v, promulgation.n, pronounce.v, pronouncement.n, proposal.n (32), propose.v (29), proposition.n (7), reaffirm.v (1), recount.v, refute.v (1), reiterate.v (1), relate.v, remark.n, remark.v, report.n (28), report.v (21), say.v (350), smirk.v, speak.v (16), state.v (4), statement.n (11), suggest.v (7), talk.v (21), tell.v (44), venture.v, write.v (18)
CHANGE_POSITION_ON_A_SCALE	accelerated.a, advance.v, balloon.v, climb.v, contract.v, contraction.n, decline.n (5), decline.v, decrease.n (3), decrease.v (2), depressed.a, depression.n, diminish.v (1), dip.v, double.v (10), down.prep (27), drop.v (6), dwindle.v, edge.v, elevated.a, elevation.n, escalation.n, explode.v, explosion.n, fall.n, fall.v (5), fluctuate.v, fluctuation.n, gain.n (3), gain.v (3), grow.v (20), growing.a (2), growth.n (25), hike.n (8), increase.n (77), increase.v (30), increasingly.adv, jump.v (2), lower.v, move.v (2), mushroom.v, plummet.v (2), reach.v (7), rise.n (11), rise.v (13), rocket.v, shift.n, shift.v, skyrocket.v (5), slide.v, soar.v (2), swell.v, swing.v, triple.v (4), tumble.n, tumble.v
CAUSATION	because of.prep (61), because.c (109), bring about.v, bring on.v, bring.v (3), causative.a, cause.n (5), cause.v (17), consequence.n (1), consequent.a, consequential.a, dictate.v, due to.prep (11), for.c, force.v (17), give rise.v, induce.v, lead (to).v (12), leave.v (19), legacy.n (1), make.v (66), mean.v (19), motivate.v, precipitate.v, put.v (41), raise.v, reason.n (16), render.v, responsible.a (11), result (in).v (15), result.n (11), resultant.a, resulting.a (2), see.v, send.v, since.c (4), so.c (31), sway.v, wreak.v
CAUSE_CHANGE_OF_POSITION_ON_A_SCALE	add.v (22), crank.v, curtail.v, cut.n (81), cut.v (61), decrease.v, development.n (10), diminish.v, double.v (10), drop.v, enhance.v, growth.n, increase.v (21), knock down.v, lift.v, lower.v (6), move.v, promote.v, push.n, push.v (6), raise.v, reduce.v (45), reduction.n (14), slash.v (9), step up.v, swell.v

Table A.2: Frequencies of frame instances per lexical unit (LU) in the corpus of factual-claim specific frames (– *continued from previous page*)

TAKING_SIDES	against.prep (21), back.v (6), backing.n, believe in.v (7), endorse.v (12), for.prep (8), in favor.prep (5), opponent.n (15), oppose.v (19), opposition [act].n (2), opposition [entity].n, part.n, pro.adv (8), side.n (6), side.v (1), support.v (81), supporter.n (15), supportive.a
CAPABILITY	ability.n (6), able.a (24), can.v (153), capability.n, capable.a (4), capacity.n (3), inability.n, incapable.a, incapacity.n, potential.a (3), potential.n (1), power.n (10), powerful.a(4), powerless.a, powerlessness.n, unable.a (2)
CREATING	assemble.v, create.v (62), form.v (7), formation.n, generate.v (7), issuance.n (1), issue.v (11), make.v, produce.v (9), production.n (9), yield.v (2)
CONDITIONAL_OCCURRENCE	as long as.scon (3), assuming.scon, if.scon (154), in case.scon (1), in the event.prep (1), provided.scon, supposing.scon, what if.scon
OCCUPY_RANK	rank.v (12), stand.v, top.a (22)
VOTE	vote.v (105), (a/the) deciding vote.n (5)
UNIQUENESS_OF_TRAIT	the only.a (34)
RECURRING_ACTION	every.prep (25)
OCCUPY_RANK_VIA_ORDINAL_NUMBERS	No. 1.a (10)
OCCUPY_RANK_VIA_SUPERLATIVES	biggest.a (27), fastest.a (4), fewest.a (4), highest.a (61), largest.a (50), longest.a (4), most.adv (53), oldest.a(2), richest.a (2), safest.a (7), smallest.a (1), worst.a (17)
RATIO	percent of. (206), out of. (13)
COMPARING_TWO_ENTITIES	than.sc (124)
COMPARING_AT_TWO_DIFFERENT_POINTS_IN_TIME	than.sc (29)
CORRELATION	every time.adv (1), whenever.c (0)
TAKING_SIDES_CONSISTENCY	change.v (3), flip-flop.v (2), shift.v (1)
RECURRENT_ACTION_IN_FREQUENCY	time.n (36)

REFERENCES

[1] X. V. Lin, R. Socher, and C. Xiong, "Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 4870–4888.

[2] L. Bennett and S. Livingston, "The disinformation order: Disruptive communication and the decline of democratic institutions," *European Journal of Communication*, vol. 33, pp. 122–139, 2018.

[3] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–36, 2017.

[4] S. Loomba, A. de Figueiredo, S. J. Piatek, K. de Graaf, and H. J. Larson, "Measuring the impact of covid-19 vaccine misinformation on vaccination intent in the uk and usa," *Nature human behaviour*, vol. 5, no. 3, pp. 337–348, 2021.

[5] 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, pp. 1–5.

[6] N. Hassan, C. Li, and M. Tremayne, "Detecting check-worthy factual claims in presidential debates," in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM)*, 2015, pp. 1835–1838.

[7] A. Patwari, D. Goldwasser, and S. Bagchi, "Tathya: A multi-classifier system for detecting check-worthy statements in political debates," in *Proceedings of the 26th ACM International Conference on Information and Knowledge Management (CIKM)*, 2017, pp. 2259–2262.

[8] I. Jaradat, P. Gencheva, A. Barrón-Cedeño, L. Màrquez, and P. Nakov, "ClaimRank: Detecting check-worthy claims in Arabic and English," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Demonstrations*, 2018, pp. 26–30.

[9] D. Jimenez and C. Li, "An empirical study on identifying sentences with salient factual statements," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[10] C. Hansen, C. Hansen, S. Alstrup, J. Grue Simonsen, and C. Lioma, "Neural check-worthiness ranking with weak supervision: Finding sentences for fact-checking," in *Companion Proceedings of the 2019 World Wide Web Conference (WWW)*, 2019, pp. 994–1000.

[11] D. R. Lab, "Factstream," https://www.factstream.co/, 2019, [Online; accessed 19-July-2021].

[12] N. Hassan, F. Arslan, C. Li, and M. Tremayne, "Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1803–1812.

[13] B. Adair, C. Li, J. Yang, and C. Yu, "Automated pop-up fact-checking: Challenges & progress," in *Computation+Journalism Symposium*, 2019, pp. 1–5.

[14] S. Shaar, N. Babulkov, G. Da San Martino, and P. Nakov, "That is a known lie: Detecting previously fact-checked claims," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 3607–3618.

[15] A. Kazemi, K. Garimella, D. Gaffney, and S. Hale, "Claim matching beyond English to scale global fact-checking," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 2021, pp. 4504–4517.

[16] T. Mihaylova, P. Nakov, L. Màrquez, A. Barrón-Cedeño, M. Mohtarami, G. Karadzhov, and J. Glass, "Fact checking in community forums," in *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018, pp. 879–886.

[17] X. Wang, C. Yu, S. Baumgartner, and F. Korn, "Relevant document discovery for fact-checking articles," in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 525–533.

[18] S. a. Miranda, D. Nogueira, A. Mendes, A. Vlachos, A. Secker, R. Garrett, J. Mitchel, and Z. Marinho, "Automated fact checking in the news room," in *Companion Proceedings of the 2019 World Wide Web Conference (WWW)*, 2019, pp. 3579–3583.

[19] S. Jiang, S. Baumgartner, A. Ittycheriah, and C. Yu, "Factoring fact-checks: Structured information extraction from fact-checking articles," in *Proceedings of The Web Conference*, 2020, pp. 1592–1603.

[20] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, "Computational fact checking from knowledge networks," *PloS one*, vol. 10, no. 6, p. e0128193, 2015.

[21] B. Shi and T. Weninger, "Discriminative predicate path mining for fact checking in knowledge graphs," *Knowledge-based systems*, vol. 104, pp. 123–133, 2016.

[22] M. H. Gad-Elrab, D. Stepanova, J. Urbani, and G. Weikum, "Exfakt: a framework for explaining facts over knowledge graphs and text," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM)*, 2019, pp. 87–95.

[23] V.-P. Huynh and P. Papotti, "A benchmark for fact checking algorithms built on knowledge bases," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, 2019, pp. 689–698.

[24] ——, "Buckle: Evaluating fact checking algorithms built on knowledge bases," *Proceedings of the VLDB Endowment*, vol. 12, no. 12, pp. 1798–1801, 2019.

[25] A. Tchechmedjiev, P. Fafalios, K. Boland, M. Gasquet, M. Zloch, B. Zapilko, S. Dietze, and K. Todorov, "Claimskg: A knowledge graph of fact-checked claims," in *International Semantic Web Conference (ISWC)*, 2019, pp. 309–324.

[26] G. Karagiannis, I. Trummer, S. Jo, S. Khandelwal, X. Wang, and C. Yu, "Mining an" anti-knowledge base" from wikipedia updates with applications to fact checking and beyond," *Proceedings of the VLDB Endowment*, vol. 13, no. 4, pp. 561–573, 2019.

[27] S. Jo, I. Trummer, W. Yu, X. Wang, C. Yu, D. Liu, and N. Mehta, "Verifying text summaries of relational data sets," in *Proceedings of the 2019 International Conference on Management of Data (SIGMOD)*, 2019, pp. 299–316.

[28] ——, "Aggchecker: A fact-checking system for text summaries of relational data sets," *Proceedings of the VLDB Endowment (PVLDB)*, vol. 12, no. 12, pp. 1938–1941, 2019.

[29] G. Karagiannis, M. Saeed, P. Papotti, and I. Trummer, "Scrutinizer: A mixed-initiative approach to large-scale, data-driven claim verification," *Proc. VLDB Endow.*, vol. 13, no. 11, pp. 2508–2521, 2020.

[30] H. Allcott, M. Gentzkow, and C. Yu, "Trends in the diffusion of misinformation on social media," *Research & Politics*, vol. 6, no. 2, p. 2053168019848554, 2019.

[31] G. Pennycook and D. G. Rand, "Who falls for fake news? the roles of bullshit receptivity, overclaiming, familiarity, and analytic thinking," *Journal of Personality*, 2019.

[32] A. J. Berinsky, "Rumors and health care reform: Experiments in political misinformation," *British Journal of Political Science*, vol. 47, no. 2, pp. 241–262, 2017.

[33] A. Bovet and H. A. Makse, "Influence of fake news in twitter during the 2016 us presidential election," *Nature Communications*, vol. 10, no. 1, 2019.

[34] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley framenet project," in *36th Annual Meeting of the Association for Computational Linguistics and 17th*

International Conference on Computational Linguistics (COLING–ACL), 1998, pp. 86–90.

[35] B. Nyhan and J. Reifler, "Estimating fact-checking's effects," 2015. [Online]. Available: https://www.americanpressinstitute.org/wp-content/uploads/2015/04/Estimating-Fact-Checkings-Effect.pdf

[36] K. Meng, D. Jimenez, F. Arslan, J. D. Devasier, D. Obembe, and C. Li, "Gradient-based adversarial training on transformer networks for detecting check-worthy factual claims," *arXiv preprint arXiv:2002.07725*, 2020.

[37] P. Nakov, A. Barrón-Cedeno, T. Elsayed, R. Suwaileh, L. Màrquez, W. Zaghouani, P. Atanasova, S. Kyuchukov, and G. Da San Martino, "Overview of the clef-2018 checkthat! lab on automatic identification and verification of political claims," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, 2018, pp. 372–387.

[38] T. Elsayed, P. Nakov, A. Barrón-Cedeno, M. Hasanain, R. Suwaileh, G. Da San Martino, and P. Atanasova, "Overview of the clef-2019 checkthat! lab: automatic identification and verification of claims," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, 2019, pp. 301–321.

[39] A. Barrón-Cedeño, T. Elsayed, P. Nakov, G. Da San Martino, M. Hasanain, R. Suwaileh, and F. Haouari, "Checkthat! at clef 2020: Enabling the automatic identification and verification of claims in social media," in *European Conference on Information Retrieval (ECIR)*, 2020.

[40] P. Nakov, G. Da San Martino, T. Elsayed, A. Barrón-Cedeno, R. Mıguez, S. Shaar, F. Alam, F. Haouari, M. Hasanain, W. Mansour, *et al.*, "Overview of the clef-2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news," in *European Conference on Information Retrieval (ECIR)*, 2021.

[41] J. Thorne and A. Vlachos, "Automated fact checking: Task formulations, methods and future directions," in *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, 2018, pp. 3346–3359.

[42] K. Lee, L. He, and L. Zettlemoyer, "Higher-order coreference resolution with coarse-to-fine inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 687–692. [Online]. Available: https://aclanthology.org/N18-2108

[43] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, and V. Raghavendra, "Deep learning for entity matching: A design space exploration," in *Proceedings of the 2018 International Conference on Management of Data (SIGMOD)*, 2018, pp. 19–34.

[44] R. Socher, E. Huang, J. Pennin, C. D. Manning, and A. Ng, "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection," *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[45] L. Han, A. L. Kashyap, T. Finin, J. Mayfield, and J. Weese, "Umbc_ebiquity-core: Semantic textual similarity systems," in *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, 2013, pp. 44–52.

[46] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 2249–2255.

[47] X. Wang, C. Yu, S. Baumgartner, and F. Korn, "Relevant document discovery for fact-checking articles," in *Proceedings of the The Web Conference*, 2018, pp. 525–533.

[48] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, "Where the truth lies: Explaining the credibility of emerging claims on the web and social media," in *Proceedings of the 26th International Conference on World Wide Web Companion (WWW)*, 2017, pp. 1003–1012.

[49] ——, "Credeye: A credibility lens for analyzing and explaining misinformation," in *Companion Proceedings of the The Web Conference 2018 (WWW)*, 2018, pp. 155–158.

[50] G. Karadzhov, P. Nakov, L. Màrquez, A. Barrón-Cedeño, and I. Koychev, "Fully automated fact checking using external sources," in *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, 2017, pp. 344–353.

[51] T. Yoneda, J. Mitchell, J. Welbl, P. Stenetorp, and S. Riedel, "UCL machine reading group: Four factor framework for fact finding (HexaF)," in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 2018, pp. 97–102.

[52] S. Tokala, G. Vishal, A. Saha, and N. Ganguly, "Attentivechecker: A bi-directional attention flow mechanism for fact verification," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2019, pp. 2218–2222.

[53] J. Zhou, X. Han, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "GEAR: Graph-based evidence aggregating and reasoning for fact verification," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019, pp. 892–901.

[54] W. Zhong, J. Xu, D. Tang, Z. Xu, N. Duan, M. Zhou, J. Wang, and J. Yin, "Reasoning over semantic-level graph for fact checking," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 6170–6180.

[55] C. Hidey, T. Chakrabarty, T. Alhindi, S. Varia, K. Krstovski, M. Diab, and S. Muresan, "DeSePtion: Dual sequence prediction and adversarial examples for improved fact-checking," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 8593–8606.

[56] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: a large-scale dataset for fact extraction and VERification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2018, pp. 809–819.

[57] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, and A. Mittal, "The fact extraction and VERification (FEVER) shared task," in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 2018, pp. 1–9.

[58] ——, "The FEVER2.0 shared task," in *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, Nov. 2019, pp. 1–6.

[59] P. Nakov, D. Corney, M. Hasanain, F. Alam, T. Elsayed, A. Barrón-Cedeño, P. Papotti, S. Shaar, and G. D. S. Martino, "Automated fact-checking for assisting human fact-checkers," *arXiv preprint arXiv:2103.07769*, 2021.

[60] Y. Nie, H. Chen, and M. Bansal, "Combining fact extraction and verification with neural semantic matching networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6859–6866.

[61] J. Thorne and A. Vlachos, "An extensible framework for verification of numerical claims," in *EACL*, 2017, pp. 37–40.

[62] F. Arslan, J. Caraballo, D. Jimenez, and C. Li, "Modeling factual claims with semantic frames," in *Proceedings of The 12th Language Resources and Evaluation Conference (LREC)*, 2020, pp. 2511–2520.

[63] F. Arslan, D. Jimenez, J. Caraballo, G. Zhang, and C. Li, "Modeling factual claims by frames," in *Proceedings of the Computation + Journalism Symposium*, 2019.

[64] K. Faasse, C. J. Chatman, and L. R. Martin, "A comparison of language use in pro- and anti-vaccination comments in response to a high profile facebook post," *Vaccine*, vol. 34, no. 47, pp. 5808–5814, 2016.

[65] J. Ruppenhofer, M. Ellsworth, M. Schwarzer-Petruck, C. R. Johnson, and J. Scheffczyk, "Framenet ii: Extended theory and practice," 2006.

[66] S. Majithia, F. Arslan, S. Lubal, D. Jimenez, P. Arora, J. Caraballo, and C. Li, "ClaimPortal: Integrated monitoring, searching, checking, and analytics of factual claims on twitter," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*, 2019, pp. 153–158.

[67] D. Shen and M. Lapata, "Using semantic roles to improve question answering," in *(EMNLP-CoNLL)*, 2007, pp. 12–21.

[68] A. Moschitti, P. Morarescu, and S. Harabagiu, "Open domain information extraction via automatic semantic labeling." in *FLAIRS Conference*, 2003, pp. 397–401.

[69] R. Sharma, A. Somani, L. Kumar, and P. Bhattacharyya, "Sentiment intensity ranking among adjectives using sentiment bearing word embeddings," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 547–552.

[70] H. C. Boas, "Bilingual FrameNet dictionaries for machine translation," in *Proceedings of the 3rd Language Resources and Evaluation Conference (LREC)*, 2002.

[71] L. Konstantinovskiy, O. Price, M. Babakar, and A. Zubiaga, "Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection," *arXiv:1809.08193*, 2018.

[72] S. Swayamdipta, S. Thomson, C. Dyer, and N. A. Smith, "Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold," *arXiv preprint arXiv:1706.09528*, 2017.

[73] L. Alasousi, S. al Hammouri, and S. al Al-abdulhadi, "Anxiety and media exposure during covid-19 outbreak in kuwait," *medRxiv*, 2020.

[74] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[75] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2019, pp. 4171–4186.

[76] P. Yin, G. Neubig, W.-t. Yih, and S. Riedel, "TaBERT: Pretraining for joint understanding of textual and tabular data," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 8413–8426.

[77] T. Yu, C.-S. Wu, X. V. Lin, Y. C. Tan, X. Yang, D. Radev, C. Xiong, *et al.*, "Grappa: Grammar-augmented pre-training for table semantic parsing," in *International Conference on Learning Representations (ICLR)*, 2021.

[78] V. Zhong, C. Xiong, and R. Socher, "Seq2sql: Generating structured queries from natural language using reinforcement learning," *arXiv preprint arXiv:1709.00103*, 2017.

[79] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, Z. Zhang, and D. Radev, "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 3911–3921.

[80] L. Dong and M. Lapata, "Coarse-to-fine decoding for neural semantic parsing," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 731–742.

[81] T. Yu, M. Yasunaga, K. Yang, R. Zhang, D. Wang, Z. Li, and D. Radev, "SyntaxSQLNet: Syntax tree networks for complex and cross-domain text-to-SQL task," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 1653–1663.

[82] J. Guo, Z. Zhan, Y. Gao, Y. Xiao, J.-G. Lou, T. Liu, and D. Zhang, "Towards complex text-to-SQL in cross-domain database with intermediate representation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019, pp. 4524–4535.

[83] X. V. Lin, R. Socher, and C. Xiong, "Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 4870–4888.

[84] D. Choi, M. C. Shin, E. Kim, and D. R. Shin, "Ryansql: Recursively applying sketch-based slot fillings for complex text-to-sql in cross-domain databases," *Computational Linguistics*, pp. 1–24, 2020.

[85] B. Wang, R. Shin, X. Liu, O. Polozov, and M. Richardson, "RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 7567–7578.

[86] O. Rubin and J. Berant, "SmBoP: Semi-autoregressive bottom-up semantic parsing," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2021, pp. 311–324.

[87] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[88] F. Li and H. V. Jagadish, "Constructing an interactive natural language interface for relational databases," *Proceedings of the VLDB Endowment*, vol. 8, no. 1, pp. 73–84, 2014.

[89] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[90] B. Adair, M. Stencel, C. Clabby, and C. Li, "The human touch in automated fact-checking: How people can help algorithms expand the production of accountability journalism," in *Computation+Journalism Symposium*, 2019.

[91] E. Spiliopoulou, E. Hovy, and T. Mitamura, "Event detection using frame-semantic parser," in *Proceedings of the Events and Stories in the News Workshop*, 2017, pp. 15–20.

[92] S. Swayamdipta, S. Thomson, C. Dyer, and N. A. Smith, "Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold," *CoRR*, vol. abs/1706.09528, 2017.

[93] F. Arslan, N. Hassan, C. Li, and M. Tremayne, "A benchmark dataset of check-worthy factual claims," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, 2020, pp. 821–829.

[94] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, V. Sable, C. Li, and M. Tremayne, "Claimbuster: The first-ever end-to-end fact-checking system," *Proceedings of the VLDB Endowment (PVLDB)*, vol. 10, no. 12, pp. 1945–1948, 2017.

[95] N. Hassan, M. Tremayne, F. Arslan, and C. Li, "Comparing automated factual claim detection against judgments of journalism organizations," in *Computation + Journalism Symposium*, 2016, pp. 1–5.