Extracting Events from Social Media and the Web

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Shi Zong, B.S, M.S.

Graduate Program in Computer Science and Engineering

The Ohio State University

2020

Dissertation Committee:

Prof. Alan Ritter, Advisor

Prof. Micha Elsner

Prof. Feng Qin

Prof. Wei Xu

Abstract

The last decade has witnessed a tremendous advance of technology and has led to an explosion of user-generated text, including the web and short informal texts in microblogs such as Twitter. It motivates the need for automatic text processing techniques to extract, aggregate and analyze this huge amount of information that no one could handle manually.

In this thesis, we present three efforts of using computational linguistics approaches to extract events from social media and the web. For each study, we develop resources and models for extracting structured information from unstructured data. We then demonstrate the value of analyzing these extracted events. In the first study, we analyze the perceived severity of cybersecurity threats reported on social media. We build a sensor that could automatically scan tweets mentioning cybersecurity threats and evaluate the threats severity based on language used to describe them. Our experimental results show that our predicted severity scores are correlated with actual scores in National Vulnerability Database (NVD) from experts and can be used as an indicator for whether a threat will be exploitable in the wild. In the second study, we study people's linguistic behavior when they make predictions about future events. We extract people's predictions from geopolitical and financial domains and investigate a number of linguistic metrics over people's justifications of their predictions. We further demonstrate the possibility of accurately predicting forecasting skills using a model that is based solely on language. In the third study, we present a corpus that could be used for automatically extracting COVID-19 related events from

Twitter. Based on our newly manually annotated dataset, we build a semantic search system that allows users to search a variety of information by using different queries related to COVID-19, such as “Who tested positive that has close contact with Boris Johnson?” or “What are the cure methods that people think effective?” We believe this semantic search system could help address the information overload for professionals who want to stay on top of recent developments related to COVID-19.

Dedicated to my family.

Acknowledgments

First and foremost, I am greatly indebted to my advisor, Alan Ritter. I could not have asked for a better advisor. Alan introduces me to the field of Natural Language Processing, especially to the track of social media and computational social science which I am very passionate about. It is Alan who teaches me how to do world-class research step by step and provides me with hands-on guidance. Alan helps develop my taste for choosing the right research topics, build my skills for designing experiments to support our claims and make me realize the importance of communicating ideas clearly to other researchers. Alan is supportive and willing to share his own experiences when I am frustrated or have difficulties in my research. He gives me the freedom to explore the topics that I am interested to fulfill my own curiosity and has faith in me that pushes me to become a better researcher. I owe Alan a lot. I wish I could become an advisor and a researcher like Alan and pass what I get from him to others.

I would like to thank my committee members: Micha Elsner, Feng Qin and Wei Xu. My interactions with them are fruitful and their insightful comments help me improve the quality of my research. A huge thanks to Wei Xu. Wei is willing to help young researchers grow. I learn a lot from her attitude towards research and it has been my privilege to work with her. I want to thank all my other collaborators: Eduard Hovy, Graham Mueller, and Evan Wright. Many thanks to all my lab mates: Ashutosh Baheti, Fan Bai, Yang Chen,

Chao Jiang, Wuwei Lan, Chaoyue Liu, Siyuan Ma, Mounica Maddela, Jeniya Tabassum, and Yang Zhong. I enjoy the open and collaborative environment that we have.

Special thanks to Branislav Kveton, who gives me the very first experience of doing research in machine learning field. During my Ph.D. study, Brano provides me with constant supports and encourages me when I face challenges. I thank my friends: Xi He, Jianyu Huangfu, Hongfu Liu, Junzhan Wang, Kangjun Wu, Hairuo Yang, Shuguan Yang, Yang Yang, Weian Yang, Yuren Yang, Yao Yao, Quan Yuan, and Zongheng Zhu. They give me unconditional tolerances and are always there whenever I need to have a chat. I hope our friendship will last forever.

Last but most importantly, I would like to thank all my family members, including my grandparents Feiyun Xiao and Danian Zhao, my parents Yiqun Zhao and Younong Zong, and my sister Chenyun Zhao, for all their unconditional love and support during my Ph.D. study. My grandmother Feiyun Xiao passed away in Sep. 2019 when I was still in United States. I hope you could see that I finally graduate with my Ph.D. degree.

Vita

2016 - present ..... Ph.D.,
Computer Science and Engineering,
The Ohio State University, USA.

2015 - 2016 ..... M.S.,
Electrical Engineering,
Carnegie Mellon University, USA.

2010 - 2014 ..... B.S.,
Electrical Engineering,
Nanjing University, China

Publications

Research Publications

Extracting COVID-19 Events from Twitter
Shi Zong, Ashutosh Baheti, Wei Xu, Alan Ritter
In *arXiv:2006.02567*

Measuring Forecasting Skill from Text
Shi Zong, Alan Ritter, Edward Hovy
In *Proceedings of the 2019 Conference of the Association for Computational Linguistics* (ACL 2020)

Analyzing the Perceived Severity of Cybersecurity Threats Reported on Social Media
Shi Zong, Alan Ritter, Graham Mueller, Evan Wright
In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics* (NAACL 2019)

Get to the Bottom: Causal Analysis for User Modeling
Shi Zong, Branislav Kveton, Shlomo Berkovsky, Azin Ashkan, Zheng Wen

In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization* (UMAP 2017)

Detecting Localized Categorical Attributes on Graphs
Siheng Chen, Yaoqing Yang, **Shi Zong**, Aarti Singh, Jelena Kovačević
In *IEEE Transactions on Signal Processing* (IEEE TSP), vol. 65, no. 10, pp. 2725-2740, 2017

Cascading Bandits for Large-Scale Recommendation Problems
Shi Zong, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, Branislav Kveton
In *Proceedings of the 32th Conference on Uncertainty in Artificial Intelligence* (UAI 2016)

Does Weather Matter? Causal Analysis of TV Logs
Shi Zong, Branislav Kveton, Sholmo Berkovsky, Azin Ashkan, Nikos Vlassis, Zheng Wen
In *Proceedings of the 26th International Conference on World Wide Web Companion* (WWW Companion 2017)

Fields of Study

Major Field: Computer Science and Engineering

Studies in:

Artificial Intelligence	Prof. Alan Ritter
Statistics	Prof. Yoonkyung Lee
Linguistic Computation	Prof. Wei Xu

Table of Contents

List of Tables

List of Figures

Figure **Page**

Chapter 1: Introduction

1.1 Motivation

We are now live in a digital age. People communicate using instant messaging apps, share their status on social media platforms like Twitter and Facebook, and express their thoughts or opinions on a variety of topics in discussion forums like Reddit. The Internet has fundamentally changed how people interact with each other and has led to an explosion of data generated from users.

Text plays a central and unique role in this big data era. A large percentage of online interactions are performed through text. More importantly, people's attitudes, opinions, and even their reasoning for making decisions are encoded in text. How to process these large and diverse collections of user-generated text, especially in automatic ways with minimal human effort, brings new challenges for computational linguistics researchers. As there are emerging demands for automatic text processing techniques from domains like biomedical or cybersecurity, it also brings challenges of how to incorporate domain knowledge, consider domain characteristics and meet domain needs when developing computational linguistic tools.

In this thesis, we focus on the Information Extraction (IE) task, using data from social media and the Web. We draw topics from cybersecurity, intelligence analysis and current COVID-19 pandemic and present our methods to automatically extract events from user-generated text. Annotated datasets and usable resources that can be used for building computational models are developed and identified for these domains. We also demonstrate that we are able to meet needs both for general public and domain experts, by analyzing these extracted signals from text. For example, one application we build can automatically generate early warnings for severe cybersecurity threats. Severe threats predicted by our model are verified to be threats that can be real exploitable. We think our presented work could help people deal with information overload by enabling them to analyze various kinds of critical information hidden in unstructured text.

1.2 Event Extraction

In this thesis, our information extraction task is formulated as extracting events with their associated attributes. For example, suppose we want to develop a system for forecasting severe cybersecurity threats, then our goal is to extract (1) objects that the threat is targeting for, and (2) the severity level of the threat. In this example, the cybersecurity threat is defined as an event, and target of the threat and the severity level are two associated terms for the threat.

In this section, we first briefly review the sentence-level event extraction literature. We then introduce our task definition. As one main focus of this thesis is to analyze the extracted events, we also make connections to computational social science.

1.2.1 Definition of Event Extraction

From a high level and a traditional point of view, sentence-level event extraction in natural language processing refers to extracting text spans that could answer the question "Who did What to Whom and Where and When".

Formally, based on the formulation of ACE 2005,[1] event extraction consists of three subtasks:

- Detecting entity mentions: Entity mentions are basically the descriptions for the events. This subtask aims at detecting the existence of an event.
- Identifying and classifying event triggers: Event triggers are verbs that express an event. Trigger identification is a binary prediction task that deciding if a given word in the text triggers an event (of any type). A followed task is to classify the identified event triggers into specific event types.
- Identifying and classifying arguments: An event argument is defined as a participant or an attribute involved in an event, following the definition by the ACE program. Identifying arguments refer to associating triggers with entities. The system then needs to classify the relationship within each entity and event trigger pair into a specific argument type, for example place and time, etc.

Our Definition. In this thesis, we take a relaxed definition of event extraction. We broadly define an event as something that happens. Event extraction then is defined as answering a sequence of questions related to this event. Questions can either ask objective or factual information, including who, what, whom, or ask subjective information like people's

[1] https://www.ldc.upenn.edu/collaborations/past-projects/ace

thoughts or opinions. This definition provides an effective and efficient way of organizing structured knowledge extracted from unstructured text, enabling further analyses of extracted information.

1.2.2 Types of Event Extraction

There are two types of event extraction in literature: domain specific event extraction and open domain event extraction. In this section, we review some representative works in two fields.

Domain specific event extraction. Domain-specific event extraction usually focuses on some specific types of events. ACE 2005 is an example of the close-domain event extraction, covering 33 event types including "life" and "transaction". Recently, there are interests in extracting events from biological domain, for example [60, 13].

Open domain event extraction. Compared to close domain event extraction, open domain event extraction does not specify the number or the type of events to be extracted beforehand. It has the advantage of requiring less annotated data, which is beneficial for tracking real-time events. For example, [94] build a system for open domain event extraction from Twitter, where each event is represented as a tuple that contains a named entity and a date. [95] consider a seed-based event extraction method for extracting cybersecurity events on Twitter.

1.2.3 Advantages of Using Social Media Data

We note two main advantages of using social media data for events extraction.

Real-time. Compared to traditional newswires, social media like Twitter provides nearly up-to-date information for what is happening around the world. It serves as a centralized

place for aggregating and organizing information from different resources. As people with all different background can post information online, Twitter also allows understanding the same event from different angles. Finally, social media data is usually easy to collect and allows large-scale user studies.

Redundancy. Another compelling reason of using Twitter data is its redundancy on information. Redundant information can be exploited to improve the performance of extraction systems [35]. Our experimental results in Chapter 5 also support this claim.

1.2.4 Connections to Computational Social Science

Computational social science is defined as the development and application of computational methods to complex, typically large-scale, human (sometimes simulated) behavioral data [67]. It is an emerging area in recent years, connecting fields like social science, psychology, economics and computational linguistics.

There have been many prior studies in computational social science field. For example, Google Flu Trends [49] use people's search queries to estimate the flu prevalence for specific regions. [5] propose a framework for distinguishing influence and homophily effects in dynamic networks. This framework can be used to understand the information diffusion in networks. Computational social science has also established a close connection with computational linguistic community. For example, [26] study the politeness from a linguistic perspective. By building models on their newly created corpus, the authors manage to show that there are statistically significant changes for Wikipedia editors before and after their elections. [117] build computational linguistic tools to study whether police officers display different levels of respect to different community members. The authors observe that officers show consistently less respect toward black versus white community members.

In [128], the authors consider how mental health counselors' linguistic change over time. It can provide guidance for more effective training for counselors. All these studies consider problems that have wide social impacts, thus could potentially benefit people by promoting equality and reducing biases.

One common starting point for computational social science studies is to prepare data and extract signals. Event extraction is one method that could enable further analyses of social science problems. For example, if our goal is to study people's opinions for topics like who will win Oscar award, then we need to first extract these opinions from unstructured data. By treating people's forecasts as an event, we could efficiently organize users' forecasts as a tuple containing attributes like users' attitudes, users' reasoning, and meta-linguistic features such as the prediction time, etc. In Chapter 4, we follow this idea and present the first study towards people's forecasting language from a linguistic perspective.

1.3 Thesis Overview

1.3.1 Our Contributions

In this thesis, we aim at addressing the following challenges. (1) We note there is a lack of high-quality annotated datasets for domain specific event extraction from user-generated text, for example for cybersecurity threats. As our goal is not only to extract factual information from text, but also try to model people's opinions or behaviors, we take the domain-specific approach and add two more event types (cybersecurity threats and COVID-19 related events) into existing literature. (2) We also note there is a limited number of NLP-based corpus analyses for understanding people's decision making process. In order to perform large-scale user study, we first extract people's forecasts or predictions

from text. It enables further linguistic analyses towards how people behave when they make predictions.

1.3.2 Thesis Outlines

The rest of this thesis is organized as follows. In Chapter 2, we provide a general overview of working with Twitter data.

In Chapter 3, we investigate methods to analyze the severity of cybersecurity threats based on the language that is used to describe them online. A corpus of 6,000 tweets describing software vulnerabilities is annotated with authors' opinions toward their severity. We show that our corpus supports the development of automatic classifiers with high precision for this task. Furthermore, we demonstrate the value of analyzing users' opinions about the severity of threats reported online as an early indicator of important software vulnerabilities. Using our predicted severity scores, we show that it is possible to accurately forecast high severity vulnerabilities and real-world exploits.

In Chapter 4, we study a specific type of people's decision-making processes: people's forecasting behaviors. We use geopolitical predictions and company earnings forecasts made by financial analysts to explore connections between the language people use to describe their predictions and their forecasting skill. We present a number of linguistic metrics which are computed over text justifications associated with people's predictions about the future including: uncertainty, readability, and emotion. By studying linguistic factors associated with predictions, we are able to shed some light on the approach taken by skilled forecasters. Furthermore, we demonstrate that it is possible to accurately predict forecasting skill using a model that is based solely on language.

In Chapter 5, we present a corpus of 10,000 tweets annotated towards COVID-19 related events, including TESTED POSITIVE, TESTED NEGATIVE, CAN NOT TEST, DEATH and CURE AND PREVENTION. We show that our corpus enables automatic identification of COVID-19 events mentioned in Twitter with text spans that fill a set of pre-defined slots for each event. Based on the extracted events, we build a semantic search system that could aggregate information, which could be beneficial for professionals to deal with information overload related to COVID-19.

Chapter 2: Collecting Data from Twitter

Traditional information extraction often starts with linguistic annotation. Twitter data has become a popular choice for reasons discussed in §1.2.3. In this chapter, we outline the typical path for working with Twitter data. The general pipeline of developing a Twitter dataset consists of the following steps: (1) data collection, (2) data pre-processing, and (3) data annotation. We discuss each step in detail below.

2.1 Data Collection

There are mainly two ways of acquiring data from Twitter. (1) The most commonly used method is to generate a list of keywords for tracking events using Twitter stream API. For example, [47] use a combination of hashtags and words to collect tweets for an initial sentiment analysis of figurative language in Twitter. [1] build a large-scale dataset for the fine-grained emotion detection task by using keywords related to emotions. (2) Based on study design, we could also first identify a list of users and then collect most recent tweets or tweets written for a given time period from these users. For example, [90] first gather a pool of active Twitter users and then collect conversations from these users. To study the language of mental health, [23] collect tweets from users who self-report their diagnoses like PTSD.

We need to pay attention to the data imbalance issue when collecting the data. It is common that a percentage of collected tweets is not relevant to our study. However, a high portion of irrelevant tweets in the collected data would cause problems for both annotations and building models. In such cases, keywords used for tracking tweets need to be adjusted to make sure a certain level of recall in the collected data. There are some other methods for solving this issue, for example [44] use a bootstrapping approach to increase the percentage of potentially offensive tweets in the collected data.

2.2 Data Pre-processing

As collected data from Twitter is usually noisy, some pre-cleaning procedures are needed before we send them for annotation and model training. Here we summarize some popular strategies of removing and filtering data. We skip the discussion for pre-processing performed on token level, for example stop words removal, URL replacement, text normalization, etc.

- Filtering based on pre-defined lexicons. For example, [79] filter the tweets using SentiWordNet [7]. [31] select tweets by using a set of keywords related to gun shoot events like "shoot" and "gun". [56] use a list of keywords and phrases to filter the dataset to only tweets related to pre-selected six issues in political discourse. [32] identify emotive tweets only if a tweet consists of at least one word derived from EMOLEX [77].
- Filtering by applying certain rules or criteria to text. For example, [76, 1] filter retweets by removing tweets started with "RT", "rt" and "Rt". [66] filter out manual retweets by using a set of pre-defined rules, for example if two tweets only differ in punctuation. [108] remove duplicated tweets by using the Jaccard similarity measure

with a threshold of 0.7. [89] only use tweets with length within 5 tokens and 200 tokens after tokenization. In [110], the authors filter tweets by putting thresholds on follower-count and time-lapse to minimize confounding effects.

- Filtering based on certain attributes. Sometimes based on the purpose of study, we need to filter tweets using attributes like users' geolocation, demographics or the number of posted tweets. For example, [41] create a dataset for geographic location prediction by only using messages tagged with physical (latitude, longitude) coordinate pairs from a mobile client. They also restrict to tweet authors who post at least 20 messages over a given time period. [36, 51] use similar filtering procedures to build geolocation prediction datasets. As most studies use only English tweets, non-English tweets could be filtered by using "lang" field in tweet or language identification packages like langid.py [50].

2.3 Data Annotations

The final step for getting supervised data is linguistic annotation. Annotations are usually performed in two formats: (1) in-house annotation and (2) crowdsourcing annotation. Here we only discuss crowdsourcing annotation.

We note the following points when collecting annotations from crowdsourcing workers. (1) Provide clear and brief instructions along with examples for workers. We notice workers normally do not spend huge time reading instructions, it is crucial that the instructions we provide are concise and to the point. (2) Use carefully designed questionnaires for collecting annotations. Crowdsourcing workers in general are not suitable for performing very complex annotation tasks, for example to choose text spans from a given text. When designing annotation tasks and interfaces, we need to take this into account and try our best

to reduce the annotation complexity for workers. If needed, the annotation task shall be divided into several phases. We need to control the number of questions and also make sure a certain percentage of questions that should be annotated with positive answers per HIT. It is helpful to make several trail runs by ourselves before releasing it to workers. (3) Take proper quality control measures during the annotation process. We need to consistently monitor the annotation progress and block workers who answer uniformly or randomly, or have a low agreement with consensus annotations or other workers. Quality measure could be done by using inter-annotator agreement metrics, including Cohen's kappa, Fleiss' kappa, and F1 score. There are some other methods for controlling annotation quality. For example, we could include quality control questions (data that has been annotated by experts) for each HIT. Increasing the number of annotations per tweet could also help improve the overall quality of the dataset.

Chapter 3: Forecasting Cybersecurity Events from Social Media

In this chapter, we discuss our methods to extract and analyze the severity of cybersecurity threats based on the language that is used to describe them online. We demonstrate that our corpus annotated towards people's perceived severity enables the development of real-time systems that continuously track cybersecurity news sources and generate alerts for new vulnerabilities before they are reported in the National Vulnerability Database (NVD).

3.1 Introduction

Software vulnerabilities are flaws in computer systems that leave users open to attack; vulnerabilities are generally unknown at the time a piece of software is first published, but are gradually identified over time. As new vulnerabilities are discovered and verified they are assigned CVE numbers (unique identifiers), and entered into the National Vulnerability Database (NVD).[2] To help prioritize response efforts, vulnerabilities in the NVD are assigned severity scores using the Common Vulnerability and Scoring System (CVSS). As the rate of discovered vulnerabilities has increased in recent years,[3] the need for efficient identification and prioritization has become more crucial. However, it is well known that a large time delay exists between the time a vulnerability is first publicly disclosed to when it

[2] https://nvd.nist.gov/

[3] https://www.cvedetails.com/browse-by-date.php

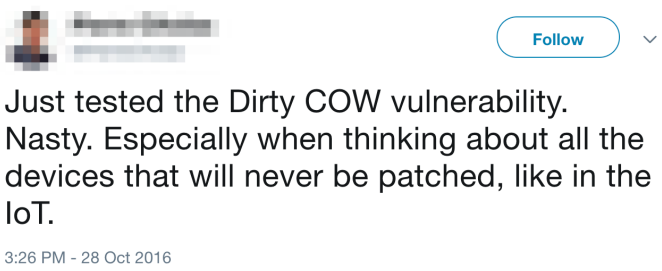


Figure 3.1: Example tweet discussing the dirty copy-on-write (COW) security vulnerability in the Linux kernel.

is published in the NVD; a recent study found that the median delay between the time a vulnerability is first reported online and the time it is published in the NVD is seven days; also, 75% of threats are first disclosed online giving attackers time to exploit the vulnerability.[4]

In this paper we present the first study of whether natural language processing techniques can be used to analyze users' opinions about the severity of software vulnerabilities reported online. We present a corpus of 6,000 tweets annotated with opinions toward threat severity, and empirically demonstrate that this dataset supports automatic classification. Furthermore, we propose a simple, yet effective method for linking software vulnerabilities reported on Twitter to entries in the NVD, using CVEs found in linked URLs. We then use our threat severity analyzer to conduct a large-scale study to validate the accuracy of users' opinions online against experts' severity ratings (CVSS scores) found in the NVD. Finally, we show that our approach can provide an early indication of vulnerabilities that result in real exploits in the wild as measured by the existence of Symantec virus signatures associated with CVEs; we also show how our approach can be used to retrospectively identify Twitter accounts that provide reliable warnings about severe vulnerabilities.

[4] https://www.recordedfuture.com/vulnerability-disclosure-delay/

Recently there has been increasing interest in developing NLP tools to identify cybersecurity events reported online, including denial of service attacks, data breaches and more [96, 19, 18]. Our proposed approach in this paper builds on this line of work by evaluating users *opinions* toward the severity of cybersecurity threats.

Prior work has also explored forecasting software vulnerabilities that will be exploited in the wild [98]. Features included structured data sources (e.g., NVD), in addition to the volume of tweets mentioning a list of 31 keywords. Rather than relying on a fixed set of keywords, we analyze message content to determine whether the author believes a vulnerability is severe. As discussed by [98], methods that rely on tracking keywords and message volume are vulnerable to adversarial attacks from Twitter bots or sockpuppet accounts [105]. In contrast, our method is somewhat less prone to such attacks; by extracting users' opinions expressed in individual tweets, we can track the provenance of information associated with our forecasts for display to an analyst, who can then determine whether or not they trust the source of information.

3.2 Analyzing Users' Opinions Toward the Severity of Cybersecurity Threats

Given a tweet t and named entity e, our goal is to predict whether or not there is a serious cybersecurity threat towards the entity based on context. For example, given the context in Figure 5.3, we aim at predicting the severity level towards *adobe flash player*. We define an author's perceived severity toward a threat using three criteria: (1) does the author believe that their followers should be worried about the threat? (2) is the vulnerability easily exploitable? and (3) could the threat affect a large number of users? If one or more of these criteria are met, then we consider the threat to be severe.

Anno. Total	1st Annotation (5 workers per tweet)			2nd Annotation (10 workers per tweet)		
	Label	# Tweets	%	Label	# Tweets	%
6,000	With threat	2,543 (1,966 for 2nd anno.)	42.4	Severe threat	506	25.7
				Moderate threat	1,460	74.3
	Without threat	3,457	57.6	/		

Table 3.1: Number of annotated tweets with break-down percentages to each category. In 1st annotation, a tweet contains a threat if more than 3 workers vote for it. In 2nd annotation, a threat is severe if more than 6 workers agree on it. Number of workers cut-offs are determined by comparing to our golden annotations in pilot studies.

3.2.1 Data Collection

To collect tweets describing cybersecurity events for annotation, we tracked the keywords "ddos" and "vulnerability" from Dec 2017 to July 2018 using the Twitter API. We then used the Twitter tagging tool described by [91] to extract named entities,[5] retaining tweets that contain at least one named entity. To cover as many linguistic variations as possible, we used Jaccard similarity with a threshold of 0.7 to identify and remove duplicated tweets with same date.[6]

3.2.2 Mechanical Turk Annotation

We paid crowd workers on Amazon Mechanical Turk to annotate our dataset. The annotation was performed in two phases; during the first phase, we asked workers to determine whether or not the tweet describes a cybersecurity threat toward a target entity, in the second phase the task is to determine whether the author of the tweet believes the threat is severe; only tweets that were judged to express a threat were annotated in the second phase. Each HIT contained 10 tweets to be annotated; workers were paid $0.20 per HIT.

[5] https://github.com/aritter/twitter_nlp

[6] We sampled a dataset of 6,000 tweets to annotate.

In pilot studies we tried combining these two annotations into a single task, but found low inter-rater agreement, especially for the threat severity judgments, motivating the need for separation of the annotation procedure into two tasks.

Figure 5.3 shows a portion of the annotation interface presented to workers during the second phase of annotation. Details of each phase are described below, and summarized in Table 3.1.

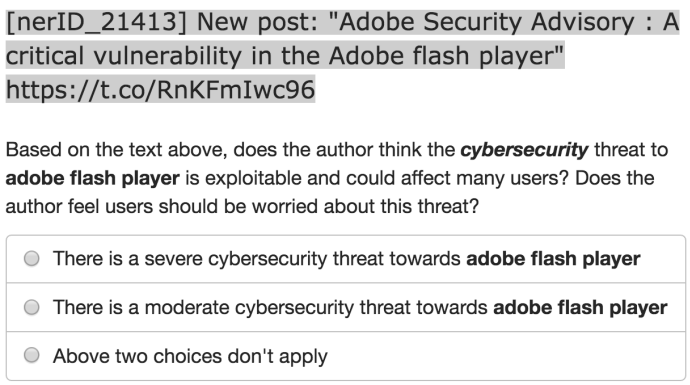


Figure 3.2: A portion of the annotation interface shown to MTurk workers during the threat severity annotation.

Threat existence annotation. Not all tweets in our dataset describe cybersecurity threats, for example many tweets discuss different senses of the word "vulnerability" (e.g., "It's OK to show vulnerability"). During the first phase of our annotation process, workers judged whether or not there appears to be a cybersecurity threat towards the target entity based on the content of the corresponding tweet. We provide workers with 3 options: the tweet indicates (a) a cybersecurity threat towards given entity, (b) a threat, but not towards the target entity, or (c) no cybersecurity threat. Each tweet is annotated by 5 workers.

Threat severity annotation. In the second phase, we collect all tweets judged to contain

threats by more than 3 workers in the first phase and annotated them for severity. 1,966 tweets were selected out of 6,000.[7] For each tweet we provided workers with 3 options: the tweet contains (a) a severe, (b) a moderate or (c) no threat toward the target entity. During our pilot study, we found this to be a more challenging annotation task, therefore we increased the number of annotators per tweet to 10 workers, which we found to improve agreement with our expert judgments.

Inter-annotator agreement. During both phases, we monitored the quality of workers' annotations using their agreement with each other. We calculated the annotation agreement of each worker against the majority vote of other workers. We manually removed data from workers who have an agreement less than 0.5, filling in missing annotations with new workers. We also manually removed data from workers who answered either uniformly or randomly for all HITs.

Agreement with expert judgments. To validate the quality of our annotated corpus we compared the workers' aggregated annotations against our own expert annotations. We independently annotated 150 randomly sampled tweets, 61 tweets of which are marked as containing severe or moderate threats. For threat existence annotation, we observe a 0.66 value of Cohen's κ [6] between the expert judgements and majority vote of 5 crowd workers. Although our threat severity annotation task may require some cybersecurity knowledge for accurate judgment, we still achieve 0.52 Cohen κ agreement by comparing majority vote from 10 workers with expert annotations.

[7]We further deduplicate pairs of tweets where the longest common subsequence covers the majority of the text contents. During deduplication all hashtags and URLs were removed and digits were replaced with 0.

3.2.3 Analyzing Perceived Threat Severity

Using the annotated corpus described in Section 3.2.2, we now develop classifiers that detect threats reported online and analyze users' opinions toward their severity. Specifically, given a named entity and tweet, $\langle e, t \rangle$, our goal is to estimate the probability the tweet describes a cybersecurity threat towards the entity, $p_{\text{threat}}(y|\langle e, t \rangle)$ and also the probability that the threat is severe, $p_{\text{severe}}(y|\langle e, t \rangle)$. In this section, we describe the details of these classifiers and evaluate their performance.

We experimented with two baselines to detect reports of cyberthreats and analyze opinions about their severity: logistic regression using bag-of-ngram features, and 1D convolutional neural networks. In the sections below we describe the input representations and details of these two models.

Logistic regression. We use logistic regression as our first baseline model for both classifiers. Input representations are bag-of-ngram features extracted from the entire tweet content. Example features are presented in Table 3.4. We use context windows of size 2, 3 and 4 to extract features. We map extracted n-grams that occur only once to a $\langle \texttt{UNK} \rangle$ token. In all our experiments, we replace named entities with a special token $\langle \texttt{TARGET} \rangle$; this helps prevent our models from biasing towards specific entities that appear in our training corpus. All digits are replaced with 0.

Convolutional neural networks. We also experimented with 1D convolutional neural networks [22, 61]. Given a tweet, the model first applies convolutional operations on input sequences with various filters of different sizes. The intermediate representations for each filter are aggregated using max-pooling over time, followed by a fully connected layer. We

choose convolution kernel sizes to be 3, 4 and 5-grams with 100 filters for each. We minimize cross-entropy loss using Adam [64]; the learning rate is set to 0.001 with a batch size of 1 and 5 epochs.

Word embeddings. We train our own cybersecurity domain word embeddings based on GloVe [86], as 39.7% of our tokens are treated as OOV words in GloVe pre-trained Twitter embeddings. We used a corpus of 609,470 cybersecurity-related tweets (described in Section 3.2.1) as our training corpus. The dimension of word embeddings is 50. Table 3.2 shows nearest neighbors for some sampled cybersecurity terms based on the learned embeddings.

During network training, we initialize word embedding layer with our own embeddings. We initialize tokens not in our trained embeddings by randomized vectors with uniform distribution from -0.01 to 0.01. We fine-tune the word embedding layer during training.

Token	Nearest Neighbors
#ddos	attacks, ddos, datacenter-insider, attack, #cyberattack
#hackers	hackers, sec_cyber, #blackberryz00, #malware, #hacking
threats	defenses, cyberrisk, #cybersecurity, threat, #iot-based
vulnerability	risk, ..., #vulnerability, strength, critical

Table 3.2: Nearest neighbors to some cybersecurity related tokens in our trained word embeddings. Embeddings are trained by using GloVe. Similar tokens are sorted by cosine similarity scores.

3.2.3.1 Experimental Setup

For threat existence classification, we randomly split our dataset of 6,000 tweets into a training set of 4,000 tweets, a development set of 1,000 tweets, and test set of 1,000 tweets. For the threat severity classifier, we only used data from 2nd phase of annotation. This dataset consists of 1,966 tweets that were judged by the mechanical turk workers to describe a cybersecurity threat towards the target entity. We randomly split this dataset into a training set of 1,200 tweets, a development set of 300 tweets, and a test set of 466 tweets. We collapsed the three annotated labels into two categories based on whether or not the author expresses an opinion that the threat towards the target entity is severe.

3.2.3.2 Results

Threat existence classifier. The logistic regression baseline has good performance at identifying threats, which we found to be a relatively easy task; area under the precision-recall curve (AUC) on the development and test set presented in Table 3.5. This enables accurate detection of trending threats online by tracking cybersecurity keywords using the Twitter streaming API, following an approach that is similar to prior work on entity-based Twitter event detection [93, 129, 122]. Table 3.3 presents an example of threats detected using this procedure on Nov. 22, 2018.[8]

Threat severity classifier. Figure 3.3 shows precision recall curves for the threat severity classifiers. Logistic regression with bag-of-ngram features provides a strong baseline for this task. Table 3.4 presents examples of high-weight features from the logistic regression model. These features often intuitively indicate severe threats, e.g. "critical vulnerability", "a massive", "million", etc. Without much hyperparameter tuning on the development set,

[8]A live demo is available at: `http://kb1.cse.ohio-state.edu:8123/events/threat`

the convolutional neural network consistently achieves higher precision at the same level of recall as compared to logistic regression. We summarize the performance of our threat existence and severity classifiers in Table 3.5.

Named Entity	Example Tweet	Existence	Severity
apple	RT AsturSec: A kernel vulnerability in Apple devices gives access to remote code execution - Packt Hub #infosec #CyberSecurity https://t....	0.96	0.59
google	RT binitamshah: Unfixed spoofing vulnerability in Google Inbox mobile apps https://t.co/TWx7jSi1gc	0.78	0.17
adobe	RT Anomali: Adobe released patches for three “important-ranked” severity vulnerabilities, including one vulnerability in Adobe Acrobat and...	0.76	0.32
flash	Vulnerability in Flash player allowing code execution. Patch before Black Friday: https://t.co/4idb570d1E #CyberSecurity #vulnerability	0.71	0.43
mac	adobe’s flash player for windows, mac and linux has a critical vulnerability that should be patched as a top priori... https://t.co/LLlPATy9vR	0.69	0.88

Table 3.3: Top five threats extracted with highest confidence on 2018/11/22. For each entity we aggregate tweets, and average threat existence scores. The tweet with the maximum threat severity score is shown in each instance.

3.3 Forecasting Severe Cybersecurity Threats

In Section 3.2 we presented methods that can accurately detect threats reported online and analyze users’ opinions about their severity. We now explore the effectiveness of this model for forecasting. Specifically, we aim to answer the following questions: **(1)** To what extent do users’ opinions about threat severity expressed online align with expert judgments? **(2)** Can these opinions provide an early indicator to help prioritize threats based on their severity?

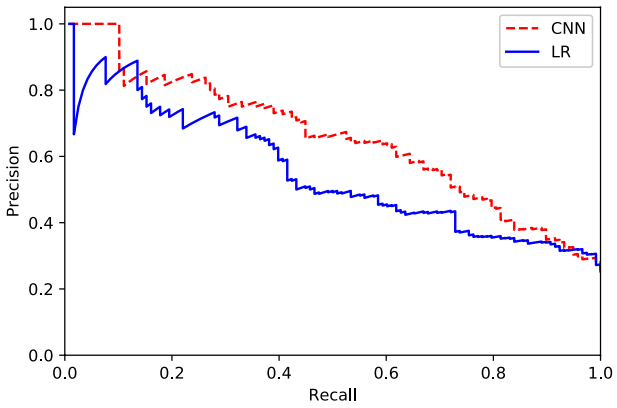


Figure 3.3: Precision/Recall curves showing performances of convolutional model (CNN) and logistic regression model (LR) for threat severity classification task in test set.

Features	Weight	Features	Weight
ddos attack	1.40	⟨TARGET⟩ ,	0.89
hackers to	1.11	take over	0.87
a massive	1.07	00 countries	0.85
critical vulnerability	1.03	attackers to	0.84
0 billion	0.96	discovered in	0.82
lets attackers	0.95	000 million	0.82
⟨TARGET⟩ users	0.91	: #ddos	0.81
a critical	0.91	abuse and	0.81
of a	0.89	, ddos	0.81
many ⟨TARGET⟩	0.89	a severe	0.79

Table 3.4: High-weight n-gram features from logistic regression model for threat severity classification task.

A large corpus of users' opinions. We follow the same procedure described in Section 3.2.1 to prepare another dataset for a large-scale evaluation. For this purpose, we collected data from Jan 2016 to Nov 2017; this ensures no tweets overlap with those that were annotated in Section 3.2.2. We collect all English tweets that explicitly contain the

Task	Model	Dev AUC	Test AUC
Existence	LR	0.88	0.85
Severity	LR	0.62	0.54
	CNN	0.70	0.65

Table 3.5: Performance of our threat existence and severity classifiers. We show area under the precision-recall curve (AUC) for both development and test sets.

keyword "vulnerability" within this time period, which results in a total number of 976,180 tweets. 377,468 tweets remain after removing tweets without named entities.

National Vulnerability Database (NVD). NVD is the U.S. government database of software vulnerabilities. Started in 2000, NVD covers over 100,000 vulnerabilities, assigning a unique CVE number for each threat. These CVE numbers serve as common identifiers. NVD uses the Common Vulnerability Scoring System (CVSS) to measure the severity of threats. CVSS currently has two versions: CVSS v2.0 and CVSS v3.0 standards. CVSS v3.0 is the latest version released in July 2015. We summarize the two standards in Table 3.6.[9]

Severity	Base Score	Severity	Base Score
		None	0.0
Low	0.0-3.9	Low	0.1-3.9
Medium	4.0-6.9	Medium	4.0-6.9
High	7.0-10.0	High	7.0-8.9
		Critical	9.0-10.0

Table 3.6: Qualitative severity rankings of vulnerabilities in NVD. (Left) CVSS v2.0 standards and (Right) CVSS v3.0 standards.

[9] https://nvd.nist.gov/vuln-metrics/cvss

Matching tweets with NVD records. Evaluating our forecasts of high severity vulnerabilities relies on accurately matching tweets describing vulnerabilities to their associated NVD records. To achieve this we present a simple, yet effective method that makes use of content in linked webpages. We find that 82.4% of tweets contain external urls in our dataset.

Our approach to link tweets to CVEs is to search for CVE numbers either in url addresses or in corresponding web pages linked in tweets reporting vulnerabilities.[10] We ignore web pages that contain more than one unique CVE to avoid potential ambiguities. Using this approach, within our dataset, 79,383 tweets were linked to 10,565 unique CVEs. In order to stimulate a forecasting scenario, we only consider CVEs where more than two associated tweets were posted at least 5 days ahead of official NVD publication date. In our dataset, 13,942 tweets are finally selected for forecast evaluation, covering 1,409 unique CVE numbers. Our matching procedure is summarized in Algorithm 1. To evaluate the accuracy of this linking procedure, we randomly sampled 100 matched pairs and manually checked them. We find the precision of our matching procedure to be very high: only 2 mismatches out of 100 are found.

3.3.1 Forecasting Models

Now that we have a linking between tweets and CVE numbers, our goal is to produce a sorted list of CVEs with those that are indicated to be severe threats the top. We consider

[10] Readers may be wondering why a CVE number has been generated before it is officially published in the database. This is due to the mechanism of assigning CVEs. Some identified companies have the right to assign CVEs or have already reversed some CVEs. When a threat appears, a CVE number is assigned immediately before any further evaluation. NVD only officially publishes a threat after all evaluations are completed. Therefore, there is a time delay between CVE entry established date and the official publication date.

[11] If a tweet or its associate urls explicitly contains a CVE number, then we ignore this maximum time range constraint.

Algorithm 1 Linking tweets to NVD records.

```
// Linking
for every tweet t do
    if CVE number in tweet context or in url links then
        match CVEs to this tweet
    else
        query webpage contents to search for CVEs

// Check linking results
Keep tweets that matched to only one unique CVE to avoid ambiguities

// Apply time constraints
Select out tweets that are posted at least 5 days ahead of official NVD publication date (at most
365 days[11])
```

two ranking procedures, detailed below; the first is based on users' opinions toward the severity of a threat, and the second is a baseline that simply uses the volume of tweets describing a specific vulnerability to measure its severity. To simplify the exposition below, we denote each CVE number as CVE_i, and the collection of tweets linked to this CVE number as $T_{\text{CVE}_i} = \{k|\text{tweet } t_k \text{ is mapped to CVE}_i\}$.

Our model: Our severe threat classifier assigns a severity score $p_{\text{severity}}(y|\langle e, t\rangle)$ for each tuple of name entity e and corresponding tweet t. For a specific CVE, we define our severity forecast score to be the maximum severity scores among all tuples from matched tweets $\langle \cdot, t_k\rangle$ (a single tweet may contain more than one name entity):

$$(\text{CVE}_i)_{\text{forecast score}} = \max_{k \in T_{\text{CVE}_i}} p_{\text{severity}}(y|\langle \cdot, t_k\rangle).$$

Tweet volume baseline: Intuitively, the number of tweets and retweets can indicate people's concern about a specific event. Specifically, the severity for threat CVE_i according to the volume model is defined by the cardinality of T_{CVE_i}:

$$(\text{CVE}_i)_{\text{volume score}} = |T_{\text{CVE}_i}|.$$

3.3.2 Forecasting CVSS Ratings

In our first set of experiments, we compare our forecasted threat severity scores against CVSS ratings from the NVD. We define a threat as being severe if its CVSS score is $\geq$ 7.0. This cut-off corresponds to qualitative severity ratings provided by CVSS (marked as HIGH or CRITICAL in Table 3.6).[12] We use the newest v3.0 scoring system, which was developed to improve v2.0.[13] Large software vendors have announced of the adaptation of the CVSS v3.0 standards, including Cisco, Oracle, SUSE Linux, and RedHat.

We evaluate our models' performance at identifying severe threats five days ahead of the NVD publication date, within their top k predictions. Table 3.7 shows our results. We observe that tweet volume performs better than a random baseline; having a large number of tweets beforehand is a good indicator for high severity, however our approach which analyzes the content of messages discussing software vulnerabilities achieves significantly better performance; 86% of its top 50 forecasts were indeed rated as HIGH or CRITICAL severity in the NVD.

	P@10	P@50	P@100	AUC
Random	59.0	61.2	58.8	0.595
Volume model	70.0	68.0	70.0	0.583
Our model	100.0	86.0	78.0	0.658

Table 3.7: Model performance of identifying severe threats (CVSS scores $\geq$ 7.0) with Precision@k and area under the precision-recall curve (AUC) metrics. For majority random baseline, we average over 10 trails.

[12]The Forum of Incident Response and Security Teams (FIRST) also provides an example guideline that recommends patching all vulnerabilities with CVSS scores $\geq$ 7.0. See `https://www.first.org/cvss/cvss-based-patch-policy.pdf`.

[13]`https://www.first.org/cvss/user-guide`

	CVE Num / Name Entity	CVE Description / Matched Tweets	CVSS Scores / Our Severity	Publish Date (# Days Ahead)
(a)	CVE-2016-0728	The join_session_keyring function in security/keys/process_keys.c in the Linux kernel before 4.4.1 mishandles object references in a certain error case, which allows local users to gain privileges or cause a denial of service (integer overflow and use-after-free) via crafted keyctl commands.	7.2 HIGH (v2.0) 7.8 HIGH (v3.0)	2016-02-08
	Android	Vulnerability in the Linux kernel could allow attackers to gain access to millions of Android devices! http://thenextweb.com/insider/2016/01/20/newly-discovered-security-flaw-could-let-hackers-control-66-of-all-android-devices/ ...	0.98	2016-01-20 (+19)
	Android	A Serious Vulnerability in the Linux Kernel Hits Millions of PCs, Servers and Android Devices http://ift.tt/1OvB4JA	0.89	2016-01-20 (+19)
	Android	Millions of PCs and Android devices are at risk from a recently discovered critical zero-day vulnerability. http://goo.gl/r95ZYZ #infosec	0.89	2016-01-20 (+19)
(b)	CVE-2017-6753	A vulnerability in Cisco WebEx browser extensions for Google Chrome and Mozilla Firefox could allow an unauthenticated, remote attacker to execute arbitrary code with the privileges of the affected browser on an affected system.	9.3 HIGH (v2.0) 8.8 HIGH (v3.0)	2017-07-25
	Cisco WebEx Extensions	The Hacker News : Critical RCE Vulnerability Found in Cisco WebEx Extensions, Again - Patch Now! http://ow.ly/gR3l30dJXlj #CDTTweets	0.98	2017-07-19 (+6)
	Cisco Systems	A critical vulnerability has been discovered in the Cisco Systems' WebEx browser extension for #Chrome and #Firefox: http://s.cgvpn.net/Zu	0.94	2017-07-18 (+7)
	Cisco WebEx Extensions	"Critical RCE Vulnerability Found in Cisco WebEx Extensions, Again - Patch Now!" via The Hacker News #security http://ift.tt/2va8Wrx	0.93	2017-07-17 (+8)

Table 3.8: Top 4 threats identified by our forecast model. Severity scores are generated by using threat severity classifier in §3.2.3.

Table 3.8 presents top 4 forecast results from our model. We observe that our model can predict accurate severity level even 19 days ahead of the official published date in NVD (Table 3.8(a), (c)).

	CVE Num / Name Entity	CVE Description / Matched Tweets	CVSS Scores / Our Severity	Publish Date (# Days Ahead)
(c)	CVE-2016-5195	Race condition in mm/gup.c in the Linux kernel 2.x through 4.x before 4.8.3 allows local users to gain privileges by leveraging incorrect handling of a copy-on-write (COW) feature to write to a read-only memory mapping, as exploited in the wild in October 2016, aka "Dirty COW."	7.2 HIGH (v2.0) 7.8 HIGH (v3.0)	2016-11-10
	Linux	Serious Dirty COW bug leaves millions of Linux users vulnerable to attack: A vulnerability discovered in the ... http://tinyurl.com/zjdp268	0.97	2016-10-22 (+19)
	Linux OS	A critical vulnerability has been discovered in all versions of the Linux OS and is being exploited in the wild http://ift.tt/2es31Xc	0.95	2016-10-25 (+16)
	Linux COW	Serious vulnerability found in the Linux COW, may have persisted for a decade. http://www.bbc.co.uk/news/technology-37728010?ocid=socialflow_twitter ... http://arstechnica.com/security/2016/10/most-serious-linux-privilege-escalation-bug-ever-is-under-active-exploit/ ...	0.82	2016-10-21 (+20)
(d)	CVE-2016-7855	Use-after-free vulnerability in Adobe Flash Player before 23.0.0.205 on Windows and OS X and before 11.2.202.643 on Linux allows remote attackers to execute arbitrary code via unspecified vectors, as exploited in the wild in October 2016.	10.0 HIGH (v2.0) 9.8 CRITICAL (v3.0)	2016-11-01
	Flash	ICYMI Critical vulnerability found in Flash, being actively exploited. Patch Flash NOW https://www.grahamcluley.com/patch-flash/	0.97	2016-10-27 (+5)
	Adobe	Adobe has released a Flash Player update to patch a critical vulnerability that malicious actors have been ex... http://bit.ly/2eaTxhO	0.95	2016-10-26 (+6)
	Adobe Flash Player	A critical vulnerability for Adobe Flash Player that allows an attacker to take control of the affected system. https://helpx.adobe.com/security/products/flash-player/apsb16-36.html ...	0.80	2016-10-27 (+5)

Table 3.9: (Table 3.8 continued) Top 4 threats identified by our forecast model. Severity scores are generated by using threat severity classifier in §3.2.3.

3.3.3 Predicting Real-World Exploits

In addition to comparing our forecasted severity scores against CVSS, as described above, we also explored several alternatives suggested by the security community to evaluate our methods: (1) Symantec's anti-virus (AV) signatures[14] and intrusion-protection (IPS) signatures,[15] in addition to (2) Exploit Database (EDB).[16]

[98] suggested Symantec's AV and IPS signatures are the best available indicator for real exploitable threats in the wild. We follow their method of explicitly querying for CVE numbers from the descriptions of signatures to generate exploited threats ground truth. Exploit Database (EDB) is an archive of public exploits and software vulnerabilities. We query EDB for all threats that have been linked into NVD.[17] In total we gathered 134 CVEs verified by Symantec and EDB to be real exploits within the 1,409 CVEs used in our forecasting evaluation.

We evaluate the number of exploited threats identified within our top ranked CVEs. Table 3.10 presents our results. We observe that 7 of top 10 threats from our model were exploited in the wild. We also observe that for the actual CVSS v3.0 scores, only 1 out of the top 10 vulnerabilities was exploited.

3.3.4 Limitations of CVSS and Real-World Exploits Ground Truth.

In Section 3.3.2 - Section 3.3.3, we compare our forecast results with (1) CVSS ratings, and (2) real exploited threats identified by Symantec signatures and Exploit Database. Each of these sources of ground truth have limitations, which we discuss below.

[14] https://www.symantec.com/security-center/a-z

[15] https://www.symantec.com/security_response/attacksignatures/

[16] https://www.exploit-db.com/

[17] http://cve.mitre.org/data/refs/refmap/source-EXPLOIT-DB.html

	Top 10		Top 50		Top 100	
	P	R	P	R	P	R
True CVSS	10.0	0.7	16.0	6.0	16.0	11.9
Volume model	60.0	4.5	22.0	8.2	19.0	14.2
Our model	70.0	5.2	28.0	10.4	21.0	15.7

Table 3.10: Model performance against real-world exploited threats identified by Symantec and Exploit-DB. "True CVSS" refers to ranking CVEs based on actual CVSS scores in NVD. This model is only for reference and can not be used in real practice, as we do not know true CVSS scores when forecasting.

CVSS ratings are widely used as standard indicators for risk measurement in practice. However, one problem of CVSS ratings is that high severity threats do not necessarily lead to real-world exploits. [3] show that only a small portion (around 2%) of reported vulnerabilities were found to be exploited in the wild. Furthermore, more than half of the threats in NVD are marked as HIGH or CRITICAL, causing a large burden on vendors to fix.[18] We also notice these CVSS scores are closely tied with specific categories of threats. For example, 85.6% of buffer errors are marked as HIGH or CRITICAL, while 72.5% of information leaks were marked as MEDIUM or LOW. All these issues post challenges on how to prioritize real exploitable threats, with the goal of reducing false positives and false negatives simultaneously. Our work provides one such additional source of information for helping to prioritize threats.

The ground truth we use for real exploited threats is still an incomplete list. For example, Linux kernel vulnerabilities are less likely to appear in Symantec signatures, as Symantec does not have a security product for Linux. Identifying real exploited threats

[18] https://www.riskbasedsecurity.com/2017/05/cvssv3-when-every-vulnerability-appears-to-be-high-priority/

is a difficult task; to the best of our knowledge, there does not exist an easy-to-access list covering all exploited threats currently.

3.3.5 Identifying Accounts that Post Reliable Warnings

Finally we perform an analysis of the reliability of individual Twitter accounts. We evaluate all accounts with more than 5 tweets exceeding 0.5 confidence score from our severity classifier. Table 3.11 presents our results. Accounts in our data whose warnings were found to have highest precision when compared against CVSS include "@securityaffairs" and "@EduardKovacs", which are known to post security related information, and both have more than 10k followers.

Account Name	# Corr / # Fcst	Acc. (%)
jburnsconsult	15 / 15	100
securityaffairs	10 / 10	100
EduardKovacs	6 / 6	100
cripperz	5 / 5	100
cipherstorm	4 / 5	80

Table 3.11: List of users with top accuracies on forecasting severe cybersecurity threats.

3.4 Additional Analysis of Results

In this section, we present further analyses of people's online behaviors when discussing cybersecurity threats on social media.

We find that the real severity of threats is predictable based on users' opinions online. We observe several repeated patterns in how people describe severe threats. We summarize some of these patterns below:

- describing severity levels (see Section 3.4.1), such as “critical”, “serious”, “highly”;
- describing the number of users or devices affected, such as “millions of ⟨TARGET⟩ devices”, “huge number of”;
- potential consequences, such as “allows hackers to”, “could allow for remote code execution”, “malware”;
- alerts or warnings, such as “please be aware”, “warning”;
- suggesting immediate actions, such as “patch now”.

3.4.1 Usage of Subjective Adjectives

We notice people rely on adjectives for describing the level of severity for threats, rather than numerical scores. These subjective adjectives form our initial impressions on these threats.

We examine subjective adjectives people use for measuring threats. We run POS tagging to extract all tokens marked as `JJ`, `JJP`, and `JJS`. We then rank subjective adjectives in Subjectivity Lexicon (SUB) [124] by log-odds ratio of their occurrences in NVD descriptions for HIGH or CRITICAL threats versus MEDIUM or LOW threats. Table 3.12 presents top ranked subjective adjectives. We observe variants people are using for severe threats, e.g. “serious”, “severe”, “malicious”, etc.

3.4.2 Temporal Analysis

We collect all CVEs having matched tweets posted at least 1 day ahead of the official NVD publication date, resulting in a set of 3,678 CVEs. Within our dataset, 84.7% of CVEs are reported within 60 days after the first disclosure on social media. We observe a median of 5 days delay in our dataset, whereas some of threats have significant longer

Adj.	Ratio	Adj.	Ratio	Adj.	Ratio
serious	2.01	aware	1.61	fast	1.39
pivotal	1.95	most	1.61	original	1.39
sure	1.95	vivid	1.61	able	1.39
free	1.95	accessible	1.39	blind	1.39
active	1.79	popular	1.39	arbitrary	1.35
intelligent	1.79	deep	1.39	high	1.30
static	1.79	black	1.39	incomplete	1.25
critical	1.67	top	1.39	malicious	1.20
severe	1.61	dangerous	1.39	wily	1.10
great	1.61	wild	1.39	evil	1.10

Table 3.12: Top ranked log-odds ratio of subjective adjectives describing severe threats (CVSS scores ≥ 7.0) versus non-severe threats (CVSS scores < 7.0). Subjective adjectives are identified by using Subjectivity Lexicon (SUB) [124].

delays. For example, CVE-2016-2123[19] (Overflow Remote Code Execution Vulnerability) first appears at Twitter on Dec. 19, 2016[20], but is published in NVD on Nov. 1, 2018. It again shows the difficulty of threat evaluation and management.

3.4.3 Error Analysis

We evaluate two types of errors with respect to forecasting high severity vulnerabilities: false positive and false negative examples. We observe that some severe threats are difficult to predict based on contents in general, such as Table 3.13(a). There is no clear clue for estimating the severity level merely on tweet contents.

We present another incorrect example extracted by our forecast system in Table 3.13(b). We notice tokens like "expose users to attack", "opens up to a raft of problems", etc. This threat does seem to be exploitable and harmful to a lot of users. However, experts mark it

[19] https://nvd.nist.gov/vuln/detail/CVE-2016-2123

[20] https://twitter.com/ryf_feed/status/810981102768758784

	CVE Num	Name Entity	Tweet	Our Score	Real Severity
(a)	CVE-2017-4984	EMC VNX1VNX2 OE	threatmeter: Vuln: EMC VNX1/VNX2 OE for File CVE-2017-4984 Remote Code Execution Vulnerability http://ift.tt/2rWXQXa	0.01	10.0 HIGH (v2.0) 9.8 CRITICAL (v3.0)
(b)	CVE-2016-1730	iPhone	A newly discovered vulnerability may expose iPhone users to attack when using a Wi-Fi hotspot - via @InfosecurityMag http://owl.li/Xw3VO	0.76	5.8 MEDIUM (v2.0) 5.4 MEDIUM (v3.0)
		iPhone	Apple iOS Flaw Enables Attacks via Hotspot: The vulnerability opens up iPhone users to a raft of problems, inc... http://bit.ly/1JqGtD9	0.45	

Table 3.13: Some examples of forecast errors made by our model. (a) False negative examples: there is no clear language clue for demonstrating the severity of threats, experts are needed for threats of this kind. (b) False positive examples: there exist some signals captured by our model for being severe threats, but actual severity might be overestimated.

as of medium severity. It might be the case that the actual severity level of some threats are overestimated by some accounts.

3.5 Related Work

There is a long history of prior work on analyzing users' opinions online [123], a large body of prior work has focused on sentiment analysis [83, 97], e.g., determining whether a message is positive or negative. In this paper we developed annotated corpora and classifiers to analyze users' opinions toward the severity of cybersecurity threats reported online, as far as we are aware this is the first work to explore this direction.

Forecasting real-world exploits is a topic of interest in the security community. For example, [14] train SVM classifiers to rank the exploitability of threats. Several studies have also predicted CVSS scores from various sources including text descriptions in NVD [52, 16].

Prior work has also explored a variety of forecasting methods that incorporate textual evidence [103], including the use of Twitter message content to forecast influenza rates

[85], predicting the propagation of social media posts based on their content [111] and forecasting election outcomes [82, 107].

3.6 Conclusion

In this paper, we presented the first study of the connections between the severity of cybersecurity threats and language that is used to describe them online. We annotate a corpus of 6,000 tweets describing software vulnerabilities with authors' opinions toward their severity, and demonstrated that our corpus supports the development of automatic classifiers with high precision for this task. Furthermore, we demonstrate the value of analyzing users' opinions about the severity of threats reported online as an early indicator of important software vulnerabilities. We presented a simple, yet effective method for linking software vulnerabilities reported in tweets to Common Vulnerabilities and Exposures (CVEs) in the National Vulnerability Database (NVD). Using our predicted severity scores, we show that it is possible to achieve a Precision@50 of 0.86 when forecasting high severity vulnerabilities, significantly outperforming a baseline that is based on tweet volume. Finally we showed how reports of severe vulnerabilities online are predictive of real-world exploits.

Chapter 4: Measuring Forecasting Skill from Text

In Chapter 3, we investigate people's predictions for the severity of cybersecurity threats. Besides this, people often make predictions about the future events, for example meteorologists tell us what the weather might look like tomorrow, financial analysts predict which companies will report favorable earnings and intelligence analysts evaluate the likelihood of future geopolitical events. In this chapter, we are interested in studying people's forecasting behavior from a linguistic perspective. We extract people's forecasts from geopolitical questions and financial domain and present the first study on the connections between the language people use to explain their predictions and their forecasting skill. Our linguistic findings from people's written justifications could potentially be useful for identifying accurate predictions or potentially skilled forecasters earlier.

More broadly, we hope our presented linguistic method could be used as a complement for social science and psychology researchers to study people's cognition processes. Compared to traditional social science studies relying on surveys and in-depth laboratory interviews, our method could be applied in a much larger scale and cheaper way. It also provides a way of measuring people's behavior in a more natural situation compared to laboratory settings.

4.1 Introduction

People often make predictions about the future, for example meteorologists tell us what the weather might look like tomorrow, financial analysts predict which companies will report favorable earnings and intelligence analysts evaluate the likelihood of future geopolitical events. An interesting question is why some individuals are significantly better forecasters [73]?

Previous work has analyzed to what degree various factors (intelligence, thinking style, knowledge of a specific topic, etc.) contribute to a person's skill. These studies have used surveys or psychological tests to measure dispositional, situational and behavioral variables [72]. Another source of information has been largely overlooked, however: the language forecasters use to justify their predictions. Recent research has demonstrated that it is possible to accurately forecast the outcome of future events by aggregating social media users' predictions and analyzing their veridicality [108], but to our knowledge, no prior work has investigated whether it might be possible to measure a forecaster's ability by analyzing their language.

In this paper, we present the first systematic study of the connection between language and forecasting ability. To do so, we analyze texts written by top forecasters (ranked by accuracy against ground truth) in two domains: geopolitical forecasts from an online prediction forum, and company earnings forecasts made by financial analysts. To shed light on the differences in approach employed by skilled and unskilled forecasters, we investigate a variety of linguistic metrics. These metrics are computed using natural language processing methods to analyze sentiment [83, 125], uncertainty [29, 99], readability, etc. In addition we make use of word lists taken from the Linguistic Inquiry and Word Count (LIWC) software [113], which is widely used in psychological research. By analyzing forecasters'

texts, we are able to provide evidence to support or refute hypotheses about factors that may influence forecasting skill. For example, we show forecasters whose justifications contain a higher proportion of uncertain statements tend to make more accurate predictions. This supports the hypothesis that more open-minded thinkers, who have a higher tolerance for ambiguity tend to make better predictions [114].

Beyond analyzing linguistic factors associated with forecasting ability, we further demonstrate that it is possible to identify skilled forecasters and accurate predictions based only on relevant text. Estimating the quality of a prediction using the forecaster's language could potentially be very beneficial. For example, this does not require access to historical predictions to evaluate past performance, so it could help to identify potentially skilled individuals sooner. Also, forecasters do not always provide an explicit estimate of their confidence, so a confidence measure derived directly from text could be very useful.

4.2 Linguistic Cues of Accurate Forecasting

In this section, we are interested in uncovering linguistic cues in people's writing that are predictive of forecasting skill. We start by analyzing texts written by forecasters to justify their predictions in a geopolitical forecasting forum. Linguistic differences between forecasters are explored by aggregating metrics across each forecaster's predictions. In §4.3, we analyze the accuracy of individual predictions using a dataset of financial analysts' forecasts towards companies' (continuous) earnings per share. By controlling for differences between analysts and companies, we are able to analyze intra-analyst differences between accurate and inaccurate forecasts.

4.2.1 Geopolitical Forecasting Data

To explore the connections between language and forecasting skill, we make use of data from Good Judgment Open,[21] an online prediction forum. Users of this website share predictions in response to a number of pre-specified questions about future events with uncertain outcomes, such as: *"Will North Korea fire another intercontinental ballistic missile before August 2019?"* Users' predictions consist of an estimated chance the event will occur (for example, 5%) in addition to an optional text justification that explains why the forecast was made. A sample is presented in Figure 4.1.

Question: Will Kim Jong Un visit Seoul before 1 October 2019?
Estimated Chance: 5%
Forecast Justification: No North Korean leader has stepped foot in Seoul since the partition of the Koreas at the end of the Korean War. . . .

Figure 4.1: A sample prediction made by a user in response to a question posted by *the Economist*.

Preprocessing. Not all predictions contain associated text justifications; in this work, we only consider predictions with justifications containing more than 10 tokens. We ran `langid.py` [70] to remove forecasts with non-English text, and further restrict our data to contain only users that made at least 5 predictions with text.

In our pilot studies, we also notice some forecasters directly quote text from outside resources (like Wikipedia, New York Times, etc.) as part of their justifications. To avoid including justifications that are mostly copied from external sources, we remove forecasts that consist of more than 50% text enclosed in quotation marks from the data.

[21] `https://www.gjopen.com/`

Dataset statistics. We collected all questions with binary answers that closed before April 9, 2019, leading to a total of 441 questions. 23,530 forecasters made 426,909 predictions. During preprocessing steps, 3,873 forecasts are identified as heavily quoted and thus removed. After removing non-English and heavily quoted forecasts, forecasts with no text justifications or justifications less than 10 tokens, in addition users with fewer than 5 predictions with text, 55,099 forecasts made by 2,284 forecasters are selected for the final dataset.

The distribution of predictions made by each forecaster is heavily skewed. 8.0% of forecasters make over 50 forecasts.[22] On average, each forecaster makes 10.3 forecasts, excluding those who made over 50 predictions. In Table 4.1, we also provide breakdown statistics for top and bottom forecasters.

4.2.2 Measuring Ground Truth

In order to build a model that can accurately classify good forecasters based on features of their language, we first need a metric to measure people's forecasting skill. For this purpose we use Brier score [15], a commonly used measure for evaluating probabilistic forecasts.[23] For questions with binary answers, it is defined as:

$$\text{Forecaster's Brier Score} = \frac{1}{N}\sum_{i=1}^{N}(f_i - o_i)^2$$

Here f_i is the forecaster's estimated probability, o_i is a binary variable indicating the final outcome of the event, and N is the total number of forecasts. Brier scores can be interpreted as the mean squared error between the forecast probability and true answer; lower

[22] In our dataset, forecasters could even make over 1,000 forecasts with justifications.

[23] Other possible scoring rules exist, for example ranking forecasters by log-likelihood. For a log-likelihood scoring rule, however, we need to adjust estimates of 1.00 and 0.00, which are not uncommon in the data, to avoid zero probability events. There are many ways this adjustment could be done and it is difficult to justify one choice over another.

scores indicate better forecasts.

Ranking forecasters. Directly comparing raw Brier scores is problematic, because users are free to choose questions they prefer, and could achieve a lower Brier score simply by selecting easier questions. To address this issue, we standardized Brier scores by subtracting the mean Brier scores and dividing by the standard deviation within questions [72].

We construct a set of balanced datasets for training and evaluating classifiers by choosing the top K and bottom K forecasters respectively. In our experiments, we vary K from 100 to 1,000; when K=1,000, the task can be interpreted roughly as classifying all $\sim$2k users into the top or bottom half of forecasters.[24]

4.2.3 Linguistic Analysis

In §4.2.2, we discussed how to measure ground-truth forecasting skill by comparing a user's predictions against ground-truth outcomes. In the following subsections, we examine a selected series of linguistic phenomenon and their connections with forecasting ability. Statistical tests are conducted using the paired bootstrap [38]. As we are performing multiple hypothesis testing, we also report results for Bonferroni-corrected significance level 0.05/30.

As discussed in §4.2.1, the distribution of forecasts per user is highly skewed. To control for this, we compute averages for each forecaster and use aggregate statistics to compare differences between the two groups at the user-level. Analyses are performed over 6,639 justifications from the top 500 forecasters and 6,040 from bottom 500.

[24]Readers may wonder if there do exist differences between top and bottom forecasters. We provide justifications for our ranking approach in Section 4.8.1.

4.2.3.1 Textual Factors

Length. We first check the average length of justifications from different groups and report our results in Table 4.1. We observe that skilled forecasters normally write significantly longer justifications with more tokens per sentence. This suggests that good forecasters tend to provide more rationale to support their predictions.

Metric	Top 500	Btm 500	p
Forecasters statistics			
# users making ≥ 50 forecasts	20	14	-
Avg. forecasts (w/o above users)	9.4	9.2	-
Length & word counts			
Avg. # tokens per user	69.1	47.0	$\uparrow\uparrow\uparrow$
% answers ≥ 100 tokens per user	18.5	8.3	$\uparrow\uparrow\uparrow$
Avg. # tokens per sentence	20.9	19.2	$\uparrow\uparrow\uparrow$

Table 4.1: Statistics of our dataset. p-values are calculated by bootstrap test. $\uparrow\uparrow\uparrow$: $p < 0.001$.

Readability. We compute two widely used metrics for readability: (1) Flesch reading ease [43] and (2) Dale-Chall formula [25]. Table 4.2 summarizes our results on average readability scores. We find good forecasters have lower readability compared to bad forecasters.

It is interesting to compare this result with the findings reported by [46], who found a negative correlation between the success of novels and their readability, and also the work of [100] who found award winning articles in academic marketing journals had higher readability. Our finding that more accurate forecasters write justifications that have lower readability suggests that skilled forecasters tend to use more complex language.

Emotion. We also analyze the sentiment reflected in forecasters' written text. Rather than analyzing sentiment orientation ("positive", "negative", or "neutral"), here we focus

Metric	p	Bonferroni
Textual Factors		
Readability		
Flesch reading ease	$\downarrow\downarrow$	
Dale-Chall	$\uparrow\uparrow\uparrow$	$*$
Emotion		
Absolute sentiment strength	$\downarrow\downarrow\downarrow$	$*$
Parts of Speech		
Cardinal	$\uparrow\uparrow\uparrow$	$*$
Noun	$\uparrow\uparrow$	
Preposition	$\uparrow\uparrow\uparrow$	$*$
Pronoun	$\downarrow\downarrow\downarrow$	$*$
1st personal pronoun	$\uparrow$	
Verb	$\downarrow\downarrow\downarrow$	$*$
Cognitive Factors		
Uncertainty		
% uncertain statements	$\uparrow\uparrow\uparrow$	$*$
Tentative (LIWC)	$\uparrow\uparrow\uparrow$	$*$
Thinking style		
% forecasts with quoted text	$\uparrow\uparrow\uparrow$	$*$
Temporal orientation		
Focus on past (LIWC)	$\uparrow\uparrow$	
Focus on present & future (LIWC)	$\downarrow\downarrow\downarrow$	$*$

Table 4.2: Comparison of various metrics computed over text written by the top 500 and bottom 500 forecasters. Good forecasters tend to exhibit more uncertainty, cite outside resources, and tend toward neutral sentiment; they also use more complex language resulting in lower readability and focus more on past events. p-values are calculated by bootstrap test. The number of arrows indicates the level of p-value, while the direction shows the relative relationship between top and bottom forecasters, $\uparrow\uparrow\uparrow$: top group is higher than bottom group with $p < 0.001$, $\uparrow\uparrow$: $p < 0.01$, $\uparrow$: $p < 0.05$. Tests that pass Bonferroni correction are marked by $*$.

on measuring sentiment *strength*. We hypothesize that skilled forecasters organize their supporting claims in a more rational way using less emotional language. Many existing sentiment analysis tools (e.g., [104]) are built on corpora such as the Stanford Sentiment

Treebank, which are composed of movie reviews or similar texts. However, justifications in our dataset focus on expressing opinions towards future uncertain events, rather than simply expressing preferences toward a movie or restaurant, leading to a significant domain mismatch. In pilot studies, we noticed many sentences that are marked as negative by the Stanford sentiment analyzer on our data do not in fact express a negative emotion. We thus use Semantic Orientation CALculator (SO-CAL), a lexicon-based model proposed by [109] which has been demonstrated to have good performance across a variety of domains. The model generates a score for each justification by adding together semantic scores of words present in the justification, with a 0 score indicating a neutral sentiment. We then take the absolute values of scores from the model and calculate averages for each group. Results in Table 4.2 show that the top 500 forecasters have a significantly lower average sentiment strength compared to bottom 500 forecasters, indicating statements from skilled forecasters tend to express neutral sentiment.

Parts of Speech. As shown in Table 4.2, we observe that top forecasters use a higher percentage of cardinal numbers and nouns, while higher numbers of verbs are associated with lower forecasting ability.[25]

We also note the bottom 500 use a higher percentage of pronouns when justifying their predictions. To investigate this difference, we further separate first person pronouns[26] from second or third person pronouns. As presented in Table 4.2, first person pronouns are used more often by the top forecasters.

[25]POS tags were obtained using Stanford CoreNLP. Nouns refer to common nouns.

[26]"I", "me", "mine", "my" and "myself".

4.2.3.2 Cognitive Factors

We now evaluate a number of factors that were found to be related to decision making processes based on prior psychological studies (e.g., [72]), that can be tested using computational tools. A number of these metrics are calculated by using the Linguistic Inquiry and Word Count (LIWC) lexicon [113], a widely used tool for psychological and social science research.

Uncertainty. To test the hypothesis that good forecasters have a greater tolerance for uncertainty and ambiguity, we employ several metrics to evaluate the degree of uncertainty reflected in their written language. We use the model proposed by [2] to estimate the proportion of uncertain statements made by each forecaster in our dataset. It is an attention based convolutional neural network model, that achieves state-of-the-art results on a Wikipedia benchmark dataset from the 2010 CoNLL shared task [42]; we use the trained parameters provided by [2]. After the model assigns an uncertainty label for each sentence, we calculate the percentage of sentences marked as uncertain. Results of this analysis are reported in Table 4.2; we observe that the top 500 forecasters make a significantly greater number of uncertain statements compared to the bottom 500, supporting the hypothesis mentioned above.

Thinking style. In §4.2.1, we discussed the issue that many forecasts contain quoted text. Although we removed posts consisting of mostly quoted text as a preprocessing step, we are interested in how people use outside resources during their decision making process. We thus calculate the portion of forecasts with quotes for the two groups. We notice skilled forecasters cite outside resources more frequently. This may indicate that skilled forecasters tend to account for more information taken from external sources when making

predictions.

Temporal orientation. We make use of the LIWC lexicon [113] to analyze the temporal orientation of forecasters' justifications. We notice good forecasters tend to focus more on past events (reflected by tokens like *"ago"* and *"talked"*); bad forecasters pay more attention to what is currently happening or potential future events (using tokens like *"now"*, *"will"*, and *"soon"*). We conjecture this is because past events can provide more reliable evidence for what is likely to happen in the future.

4.2.4 Predicting Forecasting Skill

In §4.2.3, we showed there are significant linguistic differences between justifications written by skilled and unskilled forecasters. This leads to a natural question: is it possible to automatically identify skilled forecasters based on the written text associated with their predictions? We examine this question in general terms first, then present experiments using a realistic setup for early prediction of forecasting skill in §4.2.5.

Models and features. We start with a log-linear model using bag-of-ngram features extracted from the combined answers for each forecaster. We experimented with different combinations of n-gram features from sizes 1 to 4. N-grams of size 1 and 2 have best classification accuracy. We map n-grams that occur only once to a ⟨UNK⟩ token, and replace all digits with 0. Inspired by our findings in §4.2.3, we also incorporate textual and cognition factors as features in our log-linear model.

We also experiment with convolutional neural networks [62] and BERT [33]. The 1D convolutional neural network consists of a convolution layer, a max-pooling layer, and a fully connected layer. We minimize cross entropy loss using Adam [64]; the learning rate is 0.01 with a batch size of 32. We fine-tune BERT on our dataset, using a batch size of 5

and a learning rate of 5e-6. All hyperparameters were selected using a held-out dev set.

Model performance. Results are presented in Table 4.3. As we increase the number of forecasters K, the task becomes more difficult as more forecasters are ranked in the middle. However, we observe a stable accuracy around 70%. All models consistently outperform a random baseline (50% accuracy), suggesting that the language users use to describe their predictions does indeed contain information that is predictive of forecasting ability. The n-grams with largest weights in the logistic regression model are presented in Table 4.4. We find that n-grams that seem to indicate uncertainty, including: *"it seems unlikely"*, *"seem to have"* and *"it is likely"* are among the largest positive weights.

	K	100	200	300	500	1000
LR	Bag-of-ngrams	69.5	74.2	72.5	69.2	64.8
	Textual	66.0	60.8	62.0	59.3	57.4
	Cognitive	69.0	68.0	67.3	65.5	61.0
	All above	70.5	73.5	73.3	69.8	64.7
Neural	CNN	71.5	75.0	72.0	69.6	64.0
	BERT-base	74.5	77.3	74.3	69.7	65.1

Table 4.3: Accuracy (%) on classifying skilled forecasters when choosing the top K and bottom K forecasters. For logistic regression (LR), we experiment with different sets of features: bag of $\{1, 2\}$-grams, textual factors and cognitive factors in §4.2.3, and combination of all above. For neural networks (Neural), we use convolutional neural network (CNN) and BERT-base. All results are based on 5-fold cross validation.

4.2.5 Identifying Good Forecasters Earlier

With the model developed in §4.2.4, we are now ready to answer the following question: using only their first written justification, can we foresee a forecaster's future performance?

Setup. Our goal is to rank forecasters by their performance. We first equally split all

Top15 (High-weight)	in the next / . also , / . however , / based on the / there are no / . according to / of time . / . based on / they wo n't / there is no / it seems unlikely / do n't see / it is likely / more of a / seem to have
Bottom15 (Low-weight)	will continue to / it will be / the world . / . it 's / there is a / is not a / the west . / to be on / to be the / . yes , / he 's a / there will be / in the world / will still be / . he will

Table 4.4: High and low-weight n-gram features from the logistic regression model trained to identify good forecasters (K=500 with only 3-gram features for interpretability). Positive features indicate some uncertainty (e.g., *"it is likely"*, *"seem to have"* , *"it seems unlikely"*), in addition to consideration of evidence from many sources (e.g., *"based on the"*, *". according to"*).

2,284 forecasters into two groups (top half versus bottom half) based on their standardized Brier scores. We then partition them into 60% train, 20% validation, and 20% test splits within each group. We combine all justifications for each forecaster in the training set. For forecasters in the validation and test sets, we only use their single earliest forecast.

We use forecasters' final rank sorted by averaged standardized Brier score over all forecasts as ground truth. We then compare our text-based model to the following two baselines: (1) a random baseline (50%) and (2) the standardized Brier score of the users' single earliest forecast.

Results. We calculate the proportion of good forecasters identified in the top N, ranked by our text-based model, and report results in Table 4.5. We observe that our models achieve comparable or even better performance relative to the first prediction's adjusted Brier score. Calculating Brier scores requires knowing ground-truth, while our model can evaluate the performance of a forecaster *without* waiting to know the outcome of a predicted event.

	P@10	P@50	P@100
Brier score	60	64	62
Text-based (LR)	70	70	65
Text-based (CNN)	90	68	64
Text-based (BERT-base)	80	70	67

Table 4.5: Precision@N of identifying skilled forecasters based on their first prediction.

4.3 Companies' Earnings Forecasts

In §4.2, we showed that linguistic differences exist between good and bad forecasters, and furthermore, these differences can be used to predict which forecasters will perform better. We now turn to the question of whether it is possible to identify which *individual* forecasts, made by the same person, are more likely to be correct. The Good Judgment Open data is not suitable to answer this question, because forecasts are discrete, and thus do not provide a way to rank individual predictions by accuracy beyond whether they are correct or not. Therefore, in this section, we consider numerical forecasts in the financial domain, which can be ranked by their accuracy as measured against ground truth.

In this paper, we analyze forecasts of companies' earnings per share (EPS). Earnings per share is defined as the portion of a company's profit allocated to each share of common stock. It is an important indicator of a company's ability to make profits. For our purposes, EPS also supports a cleaner experimental design as compared to stock prices, which constantly change in real time.

Data. We analyze reports from the Center for Financial Research and Analysis (CFRA).[27] These reports provide frequent updates for analysts' estimates and are also organized in

[27] https://www.cfraresearch.com/

a structured way, enabling us to accurately extract numerical forecasts and corresponding text justifications.

We collected CFRA's analyst reports from the Thomson ONE database[28] from 2014 to 2018. All notes making forecasts are extracted under the *"Analyst Research Notes and other Company News"* section. The dataset contains a total of 32,807 notes from analysts, covering 1,320 companies.

4.3.1 Measuring Ground Truth

We use a pattern-based approach (in Section 4.9.1) for extracting numerical forecasts. After removing notes without EPS estimates, 16,044 notes on 1,135 companies remain (this is after removing analysts who make fewer than 100 forecasts as discussed later in this section). We next evaluate whether the text can reflect how accurate these predictions are.

Forecast error. We measure the correctness of forecasts by absolute relative error [10, 37]. The error is defined by the absolute difference between the analyst's estimate e and corresponding actual EPS o, scaled by the actual EPS:

$$\text{Forecast Error} = \frac{|e - o|}{|o|}$$

Low forecast errors indicate accurate forecasts.[29]

Ranking individual forecasts. As our goal is to study the intra-analyst differences between accurate and inaccurate forecasts, we standardize forecast errors within each analyst by subtracting the analyst's mean forecast error and then dividing by the standard deviation. To guarantee we have a good estimate for the mean, we only include analysts who make

[28] https://www.thomsonone.com/

[29] Other methods for measuring the forecasting error have been proposed, for example to scale the relative error by stock price. We do not take this approach as stock prices are dynamically changing.

at least 100 forecasts (19 analysts are selected). We notice most forecast errors are smaller than 1, while a few forecasts are associated with very large forecasting errors.[30] Including these outliers would greatly affect our estimation for analysts' mean error. Thus, we only use the first 90% of the sorted forecast errors in this calculation.

4.3.2 Predicting Forecasting Error from Text

Our goal is to test whether linguistic differences exist between accurate and inaccurate forecasts, independently of who made the prediction, or how difficult a specific company's earnings might be to predict. To control for these factors, we standardize forecasting errors within analysts (as described in §4.3.1), and create training/dev/test splits across companies and dates.

Setting. We collect the top K and bottom K predictions and split train, dev and test sets by time range and company. All company names are randomly split into 80% train and 20% evaluation sets. We use predictions for companies in the train group that were made in 2014-2016 as our training data. The dev set and test set consist of predictions for companies in evaluation group made during the years 2017 and 2018, respectively. All hyperparameters are the same as those used in §4.2.4. When evaluating the classifier's performance, we balance the data for positive and negative categories.

Results. Table 4.6 shows the performance of our classifier on the test set. We observe our classifiers consistently achieve around 60% accuracy when varying the number of top and bottom forecasts, K.

[30]For example, one analyst estimated an EPS for Fiscal Year 2015 of Olin Corporation (OLN) as $1.63, while the actual EPS was $-0.01, a standardized forecast error of 164.

	K	1000	2000	3000	5000
LR	Bag-of-ngrams	63.9	62.5	61.9	59.3
	Linguistic	56.3	59.2	55.4	55.5
	All above	64.3	64.1	61.5	59.7
Neural	CNN	66.7	67.8	64.7	64.0
	BERT-base	70.8	66.7	65.8	64.4

Table 4.6: Accuracy (%) for classifying accurate predictions when using top K and bottom K analysts' predictions. We choose n-gram sizes to be 1 and 2. All reported results are on the test set.

4.3.3 Linguistic Analysis

We present our linguistic analysis in Table 4.7. The same set of linguistic features in §4.2.3 is applied to top 4,000 accurate and bottom 4,000 inaccurate analysts notes, excluding readability metric and quotation measure in thinking style metric. Analysts' notes are written in a professional manner, which makes readability metric not applicable. The notes do not contain many quoted text so we exclude quotation measure from the analysis. We also replace the emotion metric with a sentiment lexicon specifically tailored for financial domain and provide our discussions. The Bonferroni-corrected significance level is 0.05/15. We defer discussions to §4.4 for comparing across different domains. On average, each forecast contains 132.2 tokens with 5.5 sentences.

Financial sentiment. We make use of a lexicon developed by [69], which is specifically designed for financial domain. The ratio of positive and negative sentiment terms to total number of tokens is compared. Our results show that inaccurate forecasts use significantly more negative sentiment terms.

Metric	p	Bonferroni
Parts of Speech		
Cardinal	$\uparrow\uparrow$	
Noun	$\uparrow\uparrow$	
Verb	$\downarrow\downarrow\downarrow$	$*$
Uncertainty		
% uncertain statements	$\downarrow\downarrow$	$*$
Temporal orientation		
Focus on past (LIWC)	$\uparrow\uparrow$	$*$
Focus on present & future (LIWC)	$\downarrow\downarrow\downarrow$	$*$
Financial sentiment		
Positive	$\uparrow\uparrow$	
Negative	$\downarrow\downarrow\downarrow$	$*$

Table 4.7: Comparison of various metrics over top 4,000 accurate and bottom 4,000 inaccurate forecasts. Only hypotheses with $p < 0.05$ are reported. See §4.3.3 for detailed justifications. We follow the same notation as in Table 4.2, $\uparrow\uparrow\uparrow$: $p < 0.001$, $\uparrow\uparrow$: $p < 0.01$, $\uparrow$: $p < 0.05$.

4.4 Comparison of Findings Across Domains

In §4.2 and §4.3, we analyze the language people use when they make forecasts in geopolitical and financial domains. Specifically, these two sections reveal how language is associated with accuracy both within and across forecasters. In this section, we compare our findings from these domains.

Our studies reveal several shared characteristics of accurate forecasts from a linguistic perspective over geopolitical and financial domains (in Table 4.2 and Table 4.7). For example, we notice that skilled forecasters and accurate forecasts more frequently refer to past events. We also notice accurate predictions consistently use more nouns while unskilled forecasters use more verbs.

We also note one main difference between two domains is uncertainty metric: in Good Judgment Open dataset, we observe that more skilled forecasters employ a higher level of uncertainty; while for individual forecasts, less uncertainty seems to be better. It makes us consider the following hypothesis: within each forecaster, people are more likely to be correct when they are more certain about their judgments, while in general skilled forecasters exhibit a higher level of uncertainty. To test this hypothesis, we calculate the Spearman's ρ between the financial analysts' mean forecasting errors and their average portion of uncertain statements. Results show that these two variables are negative correlated with ρ=-0.24, which provides some support for our hypothesis, however the sample size is very small (there are only 19 analysts in the financial dataset). Also, these mean forecasting errors are not standardized by the difficulty of companies analysts are forecasting.

4.5 Related Work

Many recent studies have analyzed connections between users' language and human attributes [54, 81, 120, 112, 4]. [106] developed a tool for discourse analysis in social media and found that older individuals and females tend to use more causal explanations. Another example is work by [101], who developed automatic classifiers for temporal orientation and found important differences relating to age, gender in addition to Big Five personality traits. [39] showed that language expressed on Twitter can be predictive of community-level psychological correlates, in addition to rates of heart disease. [30] analyzed political polarization in social media and [118] examined the connections between police officers' politeness and race by analyzing language. A number of studies [28, 40, 12, 84] have examined the connection between users' language on social media and depression and alcohol use [59]. Other work has analyzed users' language to study the effect of attributes, such

as gender, in online communication [8, 121, 119]. In this work we study the relationship between people's language and their forecasting skill. To the best of our knowledge, this is the first work that presents a computational way of exploring this direction.

Our work is also closely related to prior research on predicting various phenomenon from users' language. For example [110] study the effect of wording on message propagation, [48] examine the connection between language used by politicians in campaign speeches and applause and [87] explored linguistic differences between truthful and deceptive statements. [46] show linguistic cues drawn from authors' language are strong indicators of the success of their books and [115] presented an unsupervised model to analyze the helpfulness of book reviews by analyzing their text.

There have been several studies using data from Good Judgment Open or Good Judgment Project [73]. One recent study examining the language side of this data is [102]. Their main goal is to suggest objective metrics as alternatives for subjective ratings when evaluating the quality of recommendations. To achieve this, justifications written by one group are provided as tips to another group. These justifications are then evaluated on their ability to persuade people to update their predictions, leading to *real* benefits that can be measured by objective metrics. Prior work has also studied persuasive language on crowdfunding platforms [127]. In contrast, our work focuses on directly measuring forecasting skill based on text justifications.

Finally we note that there is a long history of research on financial analysts' forecasting ability [24, 21, 68]. Most work relies on regression models to test if pre-identified factors are correlated with forecasting skill (e.g., [68, 17]). Some work has also explored the use of textual information in financial domain. For example, [65] present a study of predicting companies' risk by using financial reports. We also note a recent paper on studying

financial analysts' decision making process by using text-based features from earning calls [58]. As far as we aware, our work is the first to evaluate analysts' forecasting skill based on their language.

4.6 Limitations and Future Work

Our experiments demonstrated it is possible to analyze language to estimate people's skill at making predictions about the future. In this section we highlight several limitations of our study and ethical issues that should be considered before applying our predictive models in a real-world application. In our study, we only considered questions with binary answers; future work might explore questions with multiple-choice outcomes. Prior studies have found that people's forecasting skills can be improved through experience and training [74]. Our study does not take this into account as we do not have detailed information on the forecasters' prior experience. Finally, we have not investigated the differences in our model's outputs on different demographic groups (e.g., men versus women), so our models may contain unknown biases and should not be used to make decisions that might affect people's careers.

4.7 Conclusion

In this work, we presented the first study of connections between people's forecasting skill and language used to justify their predictions. We analyzed people's forecasts in two domains: geopolitical forecasts from an online prediction forum and a corpus of company earning forecasts made by financial analysts. We investigated a number of linguistic metrics that are related to people's cognitive processes while making predictions, including: uncertainty, readability and emotion. Our experimental results support several

findings from the psychology literature. For example, we observe that skilled forecasters are more open-minded and exhibit a higher level of uncertainty about future events. We further demonstrated that it is possible to identify skilled forecasters and accurate predictions based solely on language.

4.8 Additional Experiments on Good Judgment Open Dataset

4.8.1 Differences Between Top and Bottom Forecasters?

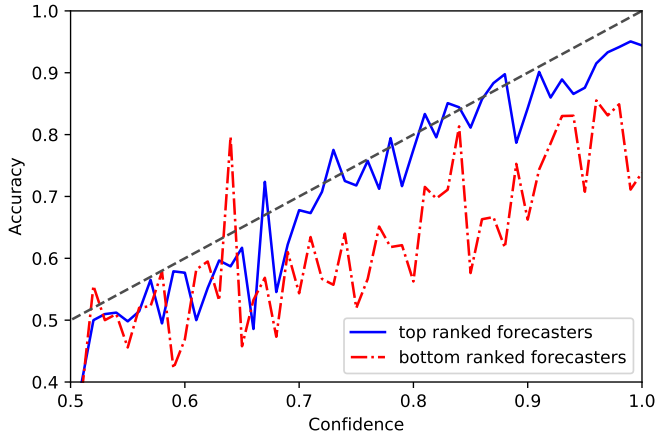
Figure 4.2 presents calibration curves and averaged standardized Brier scores across years for the top and bottom 500 forecasters. We observe the differences between these two groups are persistent over time. Controlled lab experiments from psychology have also demonstrated that top forecasters ranked by Brier scores consistently have better forecasting performance than bottom forecasters [72].

4.8.2 Additional Metrics and Examples for Linguistic Analysis

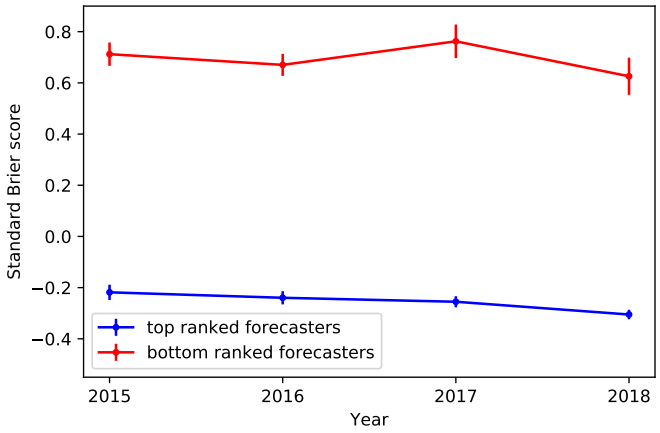
Uncertainty. We present examples of sentences with uncertainty scores from our dataset in Table 4.9.

Discourse connectives. We further investigate the portion of discourse connectives used between sentences within each group. For this purpose, we use a lexicon developed by [27], which collects connectives from PDTB corpus connective list, RST Signalling Corpus and RST-DT relational indicator list. The lexicon contains 149 English connectives, divided into 4 categories: comparison, contingency, expansion, and temporal.[31] Our results show that skilled forecasters tend to use discourse connectives more frequently compared to unskilled forecasters, which may indicate that they tend to make more coherent arguments.

[31] As some connectives are listed under more than one category, we restrict the list to those belonging to one or two categories.



(a) Calibration curves by using all forecasts.



(b) Aggregated forecasting performance across years.

Figure 4.2: Comparison of forecasting skill between the top 500 and bottom 500 forecasters ranked by averaged standardized Brier scores. (a) Calibration curves for each group calculated using all forecasts (with and without justifications). The diagonal dotted line indicates a perfect calibration. (b) Trends of average standardized Brier scores over years. Negative values indicate better forecasting skill.

Thinking style. Analytical thinking score in LIWC [113] ranks the level of a person's thinking skill. A high score correlates with formal, logical, and hierarchical thinking, while low scores are associated with informal, and narrative thinking. As shown in Table 4.8, good forecasters appear to demonstrate better analytical thinking skills.

Metric	p	Bonferroni
Discourse connectives		
Comparison	$\uparrow\uparrow\uparrow$	$*$
Contingency	$\uparrow\uparrow$	
Expansion	$\uparrow\uparrow$	$*$
Temporal	$\uparrow\uparrow\uparrow$	$*$
Thinking style		
Analytical thinking (LIWC)	$\uparrow\uparrow$	$*$

Table 4.8: Comparison of various metrics computed over text written by the top 500 and bottom 500 forecasters. p-values are calculated by bootstrap hypothesis test. The number of arrows indicates the level of p-value, while the direction shows the relative relationship between top and bottom forecasters, $\uparrow\uparrow\uparrow$: top group is higher than bottom group with $p < 0.001$, $\uparrow\uparrow$: $p < 0.01$, $\uparrow$: $p < 0.05$. Tests that pass Bonferroni correction are marked by $*$.

Sentence	Uncert. Score
Merkel is probably least prone to political scandals among the Western leaders and candidates .	1.00
It seems unlikely that the court would transfer the terms of that contract to Uber .	0.99
My assumptions : - Sturgeon will not set a date for indyref2 before the UK elections on June 8 .	0.05
To date , Toyota has distributed only 100 of the 300 Mirais preordered in California ...	0.02

Table 4.9: Examples of sentences in our dataset with uncertainty scores estimated by the model proposed by [2]. A higher uncertainty score indicates a higher level of uncertainty.

4.8.3 Linguistic Cues over Time

We are interested in whether our observed linguistic differences are consistent over time. To answer this question, we select the top 500 and bottom 500 forecasters based on their final ranking and evaluate aggregated metrics for the two groups in different years. Our results are shown in Figure 4.3. We observe the same pattern for all linguistic metrics.

For example, skilled forecasters consistently exhibit a higher level of uncertainty and past temporal orientation, and a lower readability compared to unskilled forecasters.

Sentence	We trim our 12-month target price to $20 from $23 , 10X our '16 EPS estimate of $2.01 -LRB- trimmed today from $2.10 -RRB- .
Pattern	⟨TIME⟩ EPS estimate of ⟨MONEY⟩
Extracted	⟨'16, $2.01⟩
Sentence	We raise '18 and '19 EPS estimates by $4.61 and $5.72 to $19.85 and $25.95 .
Pattern	⟨TIME⟩ and ⟨TIME⟩ EPS estimates ⟨BY−MASK⟩ to ⟨MONEY⟩ and ⟨MONEY⟩
Extracted	⟨'18, $19.85⟩, ⟨'19, $25.95⟩
Sentence	We raise our FY 17 EPS estimate to $3.23 from $2.96 and set FY 18 's at $3.43 .
Pattern	⟨TIME⟩ EPS estimate to ⟨MONEY⟩ ⟨FROM−MASK⟩ and set ⟨TIME⟩ at ⟨MONEY⟩
Extracted	⟨FY 17, $3.23⟩, ⟨FY 18, $3.43⟩

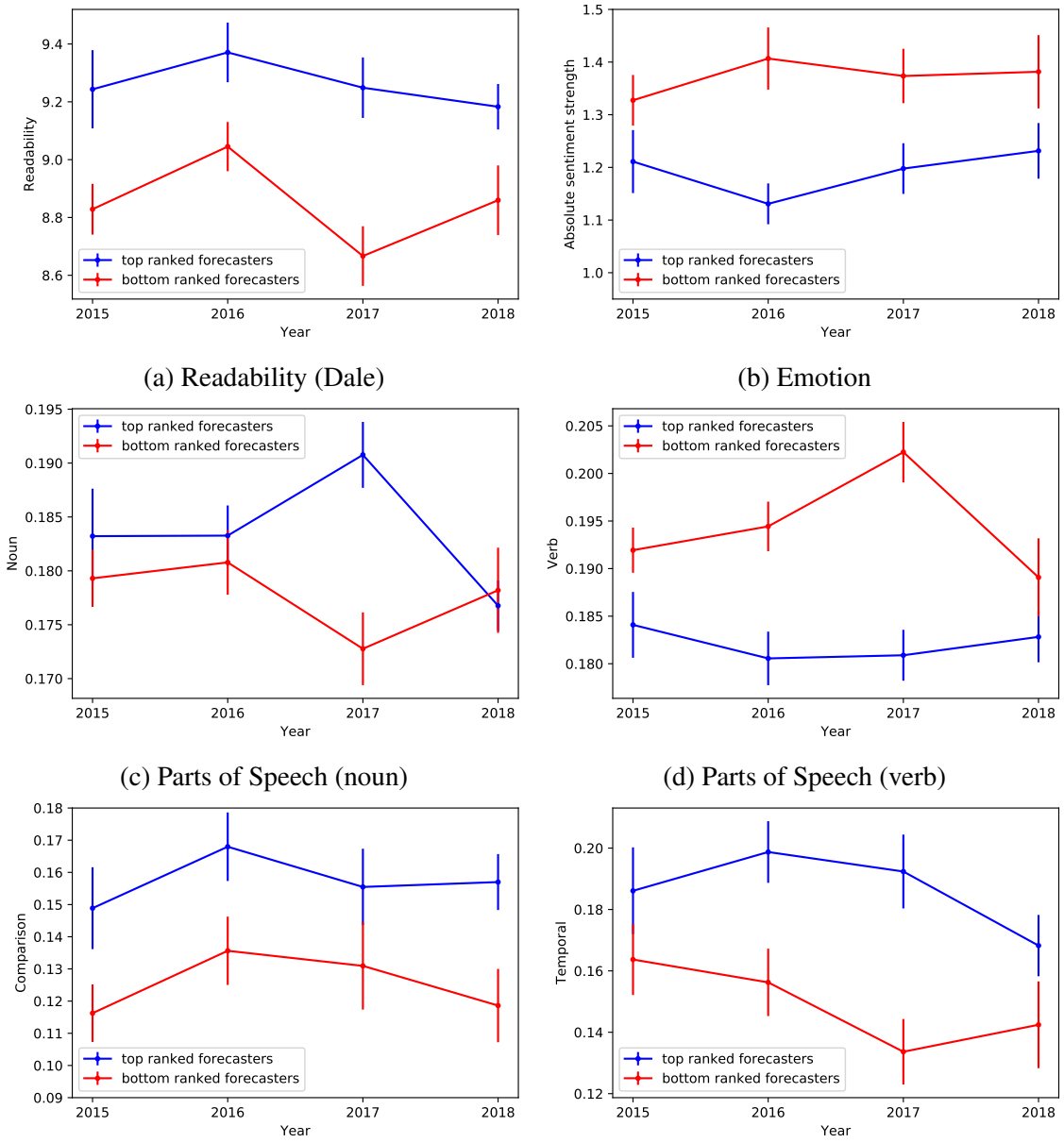
Table 4.10: Examples of earnings forecasts extracted from analysts' notes. Only sentences mentioning the earnings forecast are shown; the notes also contain additional analysis to justify the forecast. All sentences from notes are used to classify accurate versus inaccurate forecasts as described in §4.3.2.

4.9 Experimental Details on Companies' Earning Forecasts

4.9.1 Extracting Numerical Forecasts from Text

Not all analysts' notes in our dataset are associated with structured earnings forecasts (in tables). Instead, the analysts' numerical predictions for future earnings are directly reported in the text of their notes, which also contain additional language justifying their predictions. Therefore, our first goal is to extract structured representations of analysts' EPS estimates in a ⟨TIME, VALUE⟩ format. We noticed that analysts have a highly consistent style when writing this section of the report, we therefore use a set of lexico-syntactic patterns to extract the forecasts from text; as described below. We found this approach to have both high precision and high recall.

We randomly sampled 60% of the notes in our dataset for developing patterns. Before generating the rules, we replaced entities indicating time and money with special ⟨TIME⟩ and ⟨MONEY⟩ tokens. To evaluate the generalization of our patterns, we randomly sampled 100 sentences containing 136 numerical forecasts from the remaining 40% of notes and manually checked all of them. We estimate that our pattern-based approach extracts numerical forecasts with 0.91 precision and 0.82 recall. Table 4.10 shows examples of numerical forecasts extracted using our approach. In a few cases we found that an analyst's note can contain more than one forecast. For simplicity, we only consider the earliest forecast that is made within the 2014-2018 time range.



(a) Readability (Dale)

(b) Emotion

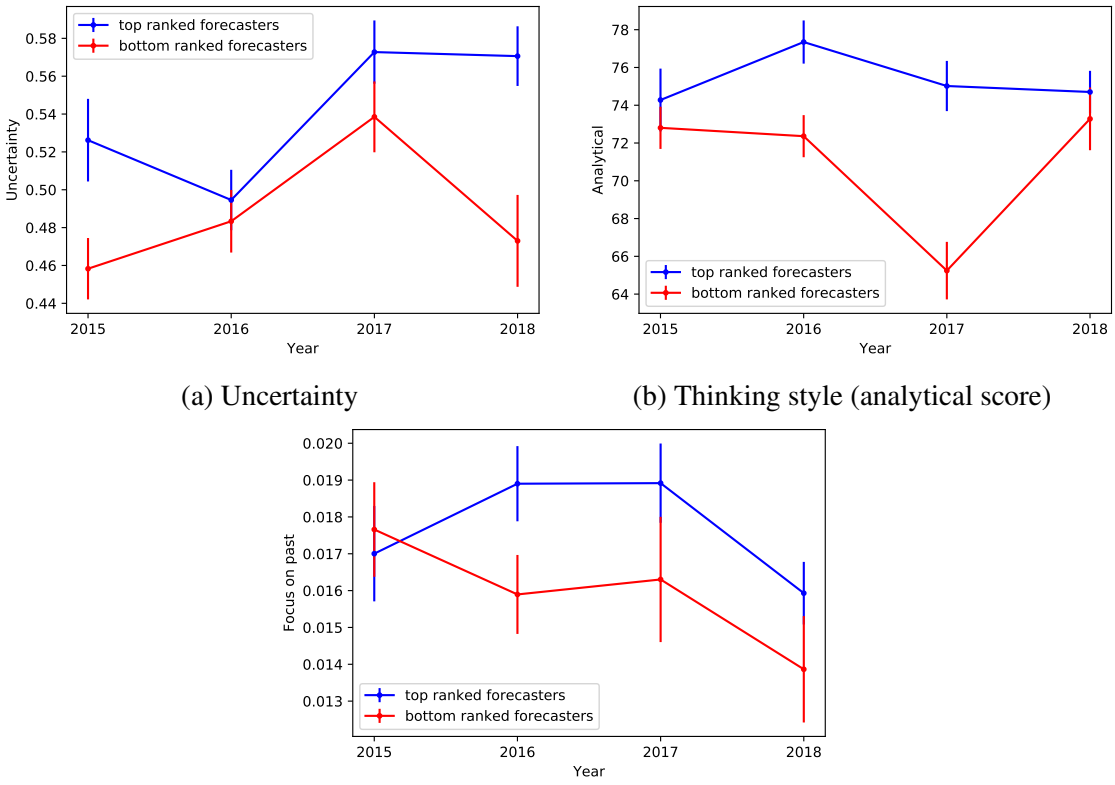
(c) Parts of Speech (noun)

(d) Parts of Speech (verb)

(e) Discourse connectives (comparison)

(f) Discourse connectives (temporal)

Figure 4.3: Linguistic features in different years for top 500 and bottom 500 forecasters. The plots show how readability (Dale), emotion, Parts of Speech (noun and verb), discourse connectives (comparison and temporal), uncertainty, thinking style (analytical score), and temporal orientation (focus on past) change in different years. We observe nearly consistent trends for all metrics over time, which indicates that linguistic differences are stable. Error bars represent standard errors.



(a) Uncertainty

(b) Thinking style (analytical score)

(c) Temporal orientation (focus on past)

Figure 4.4: (Figure 4.3 continued) Linguistic features in different years for top 500 and bottom 500 forecasters. The plots show how readability (Dale), emotion, Parts of Speech (noun and verb), discourse connectives (comparison and temporal), uncertainty, thinking style (analytical score), and temporal orientation (focus on past) change in different years. We observe nearly consistent trends for all metrics over time, which indicates that linguistic differences are stable. Error bars represent standard errors.

Chapter 5: Extracting COVID-19 Events from Twitter

In this chapter, we discuss our work on extracting COVID-19 events from Twitter. We present a new manually annotated corpus of 10,000 tweets that contain COVID-19 related events (e.g., positive/negative tests, death, denied access to test) with corresponding arguments (e.g., who, where, when). We then build automatic models trained on this corpus that can extract structured information about COVID-19 events from Twitter data. Besides, we also develop a Twitter COVID Semantic Search (TWICSS) system for users to query over the events automatically extracted from millions of tweets to demonstrate the value of COVID-19 event extraction for real-world applications. Our semantic search system is available at `http://kb1.cse.ohio-state.edu:8000/covid19`.

5.1 Introduction

Since the novel coronavirus emerged and rapidly spread across the world in December 2019, a flood of COVID-19 related information has appeared on social media. This includes reports on public figures who have tested positive/negative for the virus, which often break first on Twitter. For example, Figure 5.1 presents a tweet from U.S. presidential candidate Joe Biden who recently announced he tested negative on Twitter after a potential exposure. Besides public figures, many more ordinary Twitter users also publicly report when they have tested positive, or are experiencing symptoms but were denied access to

testing. This is a valuable source of information for people, who may want to stay informed on the latest cases reported at their work, school or other places they might visit.

In this chapter, we present the first study to extract large quantities of *structured* knowledge related to COVID-19 from Twitter automatically. To achieve this, we construct an annotated corpus of 10,000 tweets that contain five event types (i.e., positive tests, negative tests, denied access to testing, death, cure and prevention.) along with answers to slot-filling questions, such as "who tested positive?", "where did they recently travel?", "who is their employer?", etc. With this annotated corpus, we are able to train supervised learning models to extract such structured events automatically from Twitter. We find that BERT-based classifiers can achieve F_1 scores ranging from 0.4 to 0.8, depending on the event and slot.

Figure 5.1: Example tweet that contains a self-reported TESTED NEGATIVE event.

More importantly, after aggregating events extracted over a large collection of tweets, we can achieve even higher accuracy and enable users to explore the data with more complex, structured queries, such as "who tested positive that had close contact with Boris Johnson?" or "which companies in Houston have reports of employees who tested positive?". To demonstrate this, we build a semantic search system, dubbed TWICSS (Twitter

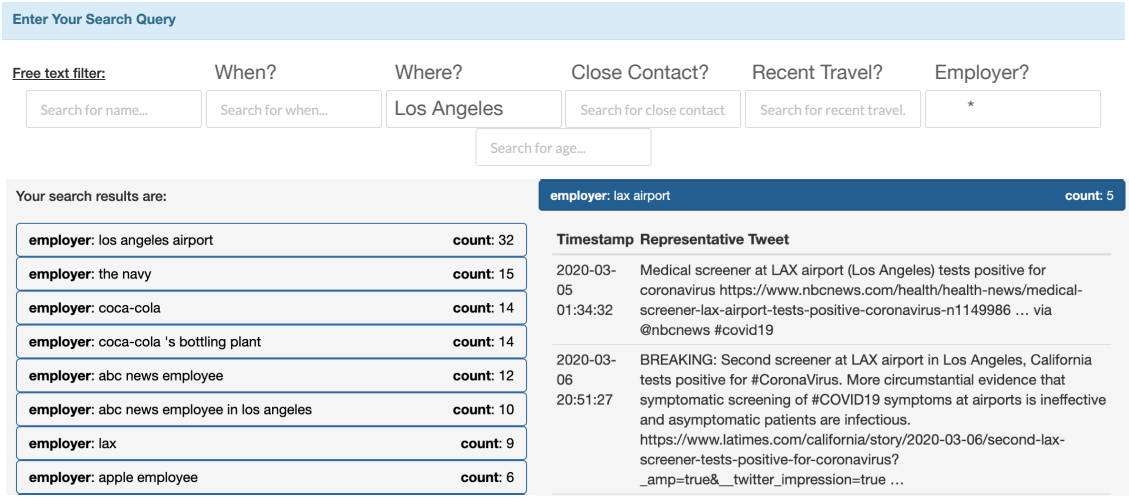


Figure 5.2: A screenshot of the user interface of our Twitter COVID Semantic Search system (TWICSS). It allows user-defined structured queries over COVID-19 events extracted from Twitter. In the example query above, the user has added a text filter, "Los Angeles", on the location slot, and indicates the results should be grouped and sorted by employer (indicated by a special token "$\star$").

COVID Semantic Search), by indexing the events automatically extracted by our model from over five million tweets using Elasticsearch.[32] We envision that TWICSS (Figure 5.2) could help to address the issue of information overload for professionals who need to stay on top of recent developments related to COVID-19, including journalists, epidemiologists and intelligence analysts.

Similar to many other applications using social media data, there are a number of important factors to consider when using COVID-related information extracted from Twitter. For example, the truthfulness of claims should be independently verified before they are assumed true. We believe that automatic event extraction could be useful for epidemiologists, journalists, or policymakers, helping them to quickly find and verify relevant pieces

[32]Elasticsearch is an open source search engine based on the Lucene library. `https://github.com/elastic/elasticsearch`

of information that are shared by users across the world. We will make all our code and data available to the research community. To protect users' privacy we will only distribute IDs of the tweets and our corresponding annotations.

5.2 Related Work

Existing COVID-19 Datasets. There have been many datasets that collect tweets related to COVID-19 [20, 9]. However, most are either unlabeled or provided with only general-purpose NLP model predictions, rather than human annotations of COVID-specific information, as in this work. For example, Twitter officially releases a COVID-19 stream with predicted entities (such as "person" and "place") and topic labels (such as "sports" and "movies"). [88] released a collection of geo-located tweets that contain COVID relevant keywords and hashtags. [34] put together 8 million tweets with predicted entities and sentiment scores. To the best of knowledge, there exist a few datasets that contain COVID related linguistic annotations at the time of writing. [53] annotated 5,000 tweets for studying the COVID-19 misconception. [80] classified 10,000 tweets into binary groups as being "informative" and "uninformative". Compared to these existing work, we provide human annotations on text spans with predefined slots for COVID-19 related event. These more fine-grained and richer annotations support training supervised learning models that are more reliable for automatic event extraction that is specific for COVID-19 applications.

Event Extraction from Twitter. There has been much interest in extracting events from Twitter. To mention just a few examples, [94] built a system for open domain event extraction from Twitter. Recent work has also explored extraction of cybersecurity events [95, 19], including denial of service attacks [18] and software vulnerabilities [131]. [130] use a nonparametric Bayesian mixture mode for event extraction. In this work, we design

event types and attributes that are specific for COVID-19 and develop automatic NLP tools for extracting structured information from tweets.

5.3 An Annotated Corpus for COVID-19 Event Extraction

To extract structured knowledge from tweets about COVID-19, we formulate the problem as a slot filling task [57, 11, 55]. That is, given a tweet, the annotators are asked to first identify whether it contains a relevant event, then mark the text spans of answers that correspond to a list of pre-defined questions for each event type. Table 5.1 shows several tweets with such annotations, while Table 5.2 shows the overall statistics of our corpus.

Tweet	**Slot Filling Annotations**
My wife's grandmother tested positive for coronavirus in an old persons home in CZ. 9 others tested positive. afaik 1 died. After 1 death they tested all residents and staff. Residents confined to rooms. They got extra staff. The grandmother has recovered now. Why is UK so bad?	**WHO** **GENDER** – F **WHERE** **RELATION** – Y
My eldest daughter tested positive for COVID-19 on Tuesday, a temperature of 40.2, she was hallucinating for hours. Now my 3 year old son looks like this. And my 5yr old girl is now showing symptoms. Still think this is a joke?!?!?!? [URL]	**WHO** **GENDER** – F **WHEN** **C. CONTACT**
apparently the staff of brikama hospital is now in isolation because a nurse tested positive for covid19...	**WHO** **EMPLOYER**
#Karnataka — A 26-year-old man returning from #Greece tested positive for #COVID19, becoming the fifth positive case in the state, a health official said on Thursday. #CoronavirusPandemic #COVID #COVID19india #CoronavirusOutbreak #coroanvirus [URL]	**WHO** **GENDER** – M **AGE** **WHERE** **RECENT V.**

Table 5.1: Examples of our annotated tweets in TESTED POSITIVE event category.

5.3.1 Data Collection

We consider five event types related to COVID: TESTED POSITIVE, TESTED NEGATIVE, CAN NOT TEST, DEATH, and CURE & PREVENTION. The first four types aim to extract structured information of events related to COVID-19, many of which are users' self-reports or news stories about public figures who have been previously exposed to the virus. We also dedicate one event type, CURE & PREVENTION, to study how some potentially misleading information are perceived by public, as there is no widely accepted antiviral treatment or vaccine for the novel coronavirus at the time of this work being conducted.

Event Type	# of Annotated Tweets	# of Slots
TESTED POSITIVE	3,000	9
TESTED NEGATIVE	1,700	8
CAN NOT TEST	1,700	5
DEATH	1,800	6
CURE & PREVENTION	1,800	3
TOTAL	10,000	31

Table 5.2: Statistics of COVID-19 Twitter Event Corpus.

We have been continuously collecting Twitter data related to COVID-19 since 2020/1/15 by tracking relevant keywords (such as "tested positive") using the Twitter API. The full list of keywords used in our data collection can be found in Table 5.4. In total, we sampled and annotated 10,000 tweets with these five event types and these corresponding slot-filling attributes. The training and validation sets consist of 7,500 annotated tweets, that are published between 2020/01/15 and 2020/04/26. To construct a balanced test set, we include 2,500 tweets published between 2020/04/27 and 2020/06/27, with 500 tweets for

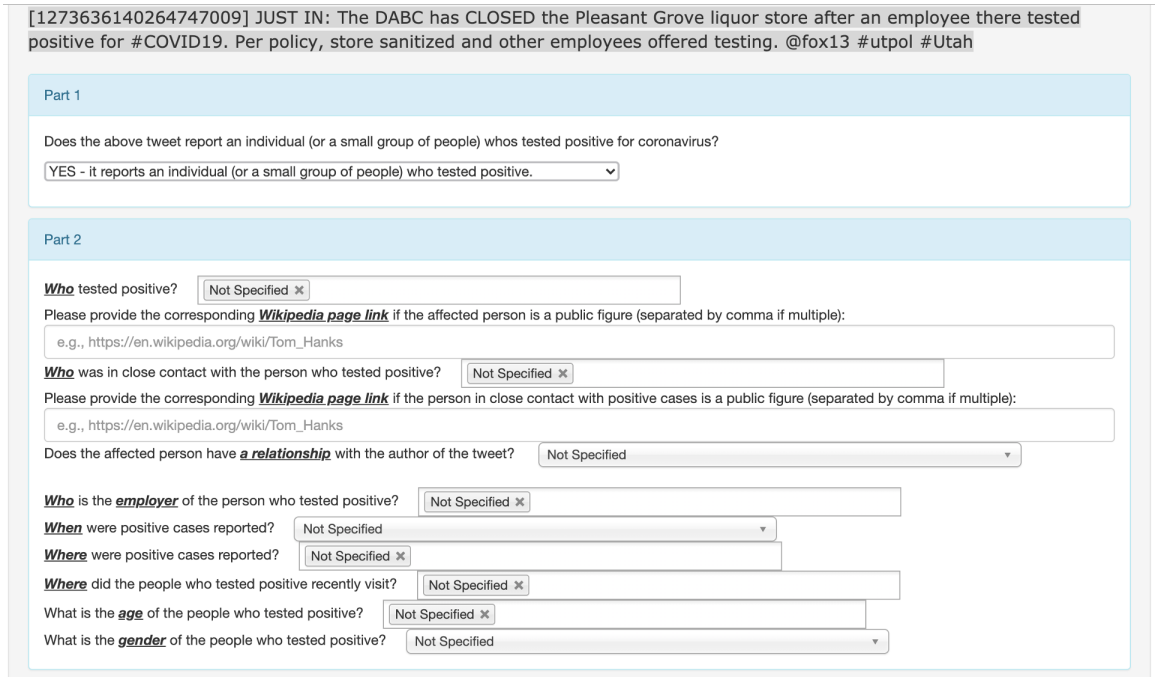


Figure 5.3: Part of the annotation interface shown to Mechanical Turk workers for TESTED POSITIVE tweets.

each event type. We have removed retweets and other duplicated tweets by keeping only one that posted the earliest. We also use Jaccard similarity with a threshold of 0.7 to remove near-identical tweets that are posted same-day.

5.3.2 Annotation Process

We hire crowd workers on Amazon Mechanical Turk to annotate our full dataset, and an in-house annotator to annotate the test set over the crowdsourced labels for another pass to further ensure the quality. Figure 5.3 shows part of the annotation interface we designed. Each tweet is annotated by 7 workers in two steps:

1. **Specific Events.** Although tweets have been filtered by keywords for each event type, many of them are generic news reports, such as, "37% of those tested under 17 for

Event Type	Slot Name	Slot Filling Questions
TESTED POSITIVE —— TESTED NEGATIVE	who	Who tested positive (negative)?
	c. contact	Who was in close contact with the person who tested positive (negative)?
	relation	Does the affected person have a relationship with the author of the tweet?
	employer	Who is the employer of the person who tested positive?
	recent v.	Where did the people who tested positive recently visit?
	when / where	{When, Where} were positive (negative) cases reported?
	age / gender	What is the {age, gender} of the people who tested positive (negative)?
	duration	How long did it take to know the result of the test?
CAN NOT TEST	who	Who can not get a test?
	relation	Does the untested person have a relationship with the author of the tweet?
	when / where	{When, Where} was the person unable to obtain a test?
	symptoms	Is the affected person currently experiencing any COVID-19 related symptoms?
DEATH	who	Who died from COVID-19?
	relation	Does the deceased person have a personal relationship with the author of the tweet?
	when / where	{When, Where} was the death reported?
	age	What is the age of the person who died?
	symptoms	Did the deceased person experience COVID-19 related symptoms?
CURE & PREVENTION	opinion	Does the author of tweet believe cure/prevention is effective?
	what	Which method of cure/prevention is mentioned?
	who	Who is promoting the cure or prevention?

Table 5.3: Slot filling questions used for collecting structured information for COVID-19 related events.

Event Type	Start From	Keywords
TESTED POSITIVE	2020/01/15	(test OR tests OR tested) positive AND VIRUS
TESTED NEGATIVE	2020/02/15	(test OR tests OR tested) negative AND VIRUS
CAN NOT TEST	2020/01/15	(can't OR can not) get (tested OR test OR tests) (can't OR can not) be tested (couldn't OR could not) get (tested OR test OR tests) (couldn't OR could not) be tested
DEATH	2020/02/15	(died OR pass away OR passed away) AND VIRUS
CURE & PREVENTION	2020/03/01	(cure OR prevent) AND VIRUS

Table 5.4: Keywords used for each event type. For VIRUS, we consider the following variants: VIRUS = (COVID19 OR COVID-19 OR corona OR coronavirus).

Coronavirus in California tested positive." Since we are interested in capturing tweets with more detailed information, we first ask the annotators to judge whether a tweet contains more specific information. For example, for tweets about positive tests, we ask the annotators whether a tweet is about an individual or a small group of people being tested positive. Annotators proceed to the next step only if they answer yes to this question.

2. **Slot Filling.** In this second step, we ask a set of pre-defined questions designed for each event type, as listed in Table 5.3. The annotators are provided with candidate answers, as a drop-down list, that include all noun phrases and named entities extracted by a Twitter-specific NLP tool [92].[33] We also combine noun phrases if they are adjacent or only separated by a preposition.[34] We include "author of the tweet" as an additional option for *who* questions and "near author of tweet" for *where* questions. For each tweet, annotators have an average of 10 to 11 possible answers to choose from, and are allowed to choose more than one answer for *WH*-questions. We also collect Wikipedia links for public figures involved in the events for potential usages in the future work.

5.3.3 Annotation Agreement

During annotation, we track crowd workers' performance by comparing their annotations with the majority vote of other workers and remove workers whose F_1 scores fall below 0.65. For the first step of annotation on specificity, the inter-annotator agreement

[33] `https://github.com/aritter/twitter_nlp`

[34] We notice in some cases these noun phrases are not perfect and may include extra words. We provide explicit guidelines for annotators stating that a candidate answer should only be chosen when it contains no more than three extra words.

between crowdsourcing workers is 0.60, measured by Cohen's kappa [6]. As for the slot-filling annotation, we compare crowdsourcing workers with an experienced in-house annotator who annotates the test set over the crowdsourced labels. Similar to previous reports on linguistic annotations on relation and event, such as ACE 2005 [75], we find that individual annotator does miss some examples. However, by aggregating annotations from multiple crowdsourcing workers,[35] we observe high agreement (an average of 0.72 F_1 score) with our in-house annotator. We also ask the in-house annotator to examine a sample of tweets where the answer span is not identified as a candidate by the automatic tagger. This scenario only occurs to less than 2% of tweets in our dataset.

5.4 Automatic Event Extraction

To demonstrate the utility of our annotated corpus, we use it to train and evaluate machine learning models for COVID-19 event extraction. Each slot filling question is treated as a binary classification problem: given a tweet t and the candidate chunk, c, the classification model $f_{(e,s)}(t, c) \rightarrow \{0, 1\}$ predicts whether c correctly answers the question for slot s, associated with event e.

5.4.1 Baseline Models

We establish two baseline methods for automatic COVID-19 event extraction:

1. **LR model.** We implement a logistic regression classifier, using bag-of-ngram features as a baseline, using (n = 1, 2, 3) for ngram features. We replace the target chunk c in the tweet with a special token <TARGET> before computing n-grams.

[35] We consider to include a span annotation for slot-filling task if 3 out of 7 MTurk annotators agree.

2. **Fine-tuning BERT model.** We fine-tune a BERT based classifier [33] that takes a tweet t as input and encloses the candidate phrase c, within the tweet, inside special entity start <E> and end </E> markers. The BERT hidden representation of token <E> is then fed as input to a linear layer to produce a binary prediction.

By design, many slots within an event are semantically related. For example, the "gender" slot is directly related to the "who" slot. During development, we found it beneficial to train the final linear layers of all slots for a given event using shared BERT parameters. We use the $\text{BERT}_{\text{base}}$ cased model and HuggingFace PyTorch implementation [126]. All shared BERT models are fine-tuned with a 2×10^{-5} learning rate using Adam [63] for 8 epochs. This model has about 110M parameters.[36]

5.4.2 Evaluation

We evaluate our model's performance on COVID-19 slot filling using held-out data from a later time period.[37] To simulate a more realistic scenario, the train, dev and test sets are partitioned by date. Both the train and dev data are taken from tweets written between 2020/01/15 to 2020/04/26; the test set consists of 500 tweets per category written between 2020/04/27 and 2020/06/27. The total number of tweets in each category is listed in Table 5.2.

Table 5.5 presents performance of the BERT and Logistic Regression models, as measured by precision (P), recall (R) and F_1 metrics.[38] We observe that our BERT-based extraction model achieves F_1 scores ranging from 0.4 to 0.8 depending on the slot, significantly

[36]The BERT for each slot is trained on a single GeForce RTX 2080 Ti GPU with average training time per epoch $\approx$ 2 minutes.

[37]The event identification task can also be solved by the slot-filling task: an event is identified if text spans are extracted for any of the pre-defined slots by our models.

[38] We excluded slots that have less than 20 annotations in the test set from evaluation, such as DURATION for TESTED NEGATIVE events and WHEN for CAN NOT TEST category.

TESTED POSITIVE		Bag-of-ngrams			BERT		
slot	#	**P**	**R**	**F$_1$**	**P**	**R**	**F$_1$**
who	375	.49	.47	.48	.84	.78	.81
c. contact	61	.33	.01	.02	.50	.34	.41
relation	21	0.0	0.0	0.0	.83	.48	.61
employer	121	.29	.10	.15	.71	.31	.43
recent v.	27	0.0	0.0	0.0	.46	.41	.43
when	22	.10	.03	.05	.78	.32	.45
where	176	.26	.27	.27	.70	.55	.61
gender m.	85	.34	.27	.30	.92	.56	.70
gender f.	31	0.0	0.0	0.0	.79	.74	.77
TESTED NEGATIVE		Bag-of-ngrams			BERT		
slot	#	**P**	**R**	**F$_1$**	**P**	**R**	**F$_1$**
who	274	.25	.21	.23	.65	.56	.60
c. contact	27	0.0	0.0	0.0	.12	.22	.15
relation	56	0.0	0.0	0.0	.65	.30	.41
where	49	0.0	0.0	0.0	.37	.45	.41
gender m.	84	.17	.09	.12	.65	.37	.47
gender f.	42	0.0	0.0	0.0	.58	.50	.54
when	27	0.0	0.0	0.0	.29	.26	.27
CAN NOT TEST		Bag-of-ngrams			BERT		
slot	#	**P**	**R**	**F$_1$**	**P**	**R**	**F$_1$**
who	153	.14	.17	.16	.59	.54	.57
relation	70	.10	.07	.08	.53	.23	.32
symptoms	52	.08	.04	.06	.38	.44	.41
where	30	.29	.15	.20	.33	.43	.37

Table 5.5: Slot-filling results on the test set for logistic regression (Bag-of-ngrams) and BERT-based classifiers. P, R and F_1 are the precision, recall and F_1 score. # is the count of gold annotations in the test data for each slot type. The last two rows report the test and dev micro-average F_1 score of classifiers for all 26 slot types combined.

outperforming the bag-of-ngrams baseline. The performance of our BERT model is sufficient to support the development of a semantic search system, which greatly benefits from redundancy of information on Twitter as discussed in §5.5.

DEATH		Bag-of-ngrams			BERT		
slot	#	**P**	**R**	**F_1**	**P**	**R**	**F_1**
who	139	.24	.37	.29	.62	.54	.58
relation	37	0.0	0.0	0.0	1.0	.24	.39
when	33	.56	.17	.26	.64	.76	.69
where	65	.29	.19	.22	.64	.42	.50
age	33	.27	.14	.18	.84	.79	.81
CURE & PREV.		Bag-of-ngrams			BERT		
slot	#	**P**	**R**	**F_1**	**P**	**R**	**F_1**
opinion	152	.40	.05	.08	.69	.55	.62
what	261	.27	.19	.22	.85	.45	.59
who	235	.27	.04	.08	.75	.29	.41
micro avg. F_1		.25			**.57**		
(dev) micro avg. F_1		.30			**.63**		

Table 5.6: (Table 5.5 continued.) Slot-filling results on the test set for logistic regression (Bag-of-ngrams) and BERT-based classifiers. P, R and F_1 are the precision, recall and F_1 score. # is the count of gold annotations in the test data for each slot type. The last two rows report the test and dev micro-average F_1 score of classifiers for all 26 slot types combined.

5.5 Semantic Search

In §5.4, we built models that can extract structured information related to COVID-19 from tweets. To demonstrate the value of our annotated dataset and models, we now describe and evaluate our semantic search system, TWICSS, that enables structured search over COVID-19 events automatically extracted from Twitter in real-time.

5.5.1 System Overview

TWICSS has a simple structured query interface that supports two operators, SELECT and GROUPBY. To construct a query, a user simply fills one or more text-filters, as illustrated in Figure 5.2. A single field is then chosen for the GROUPBY operator, as indicated by a special token, "$\star$". The system returns a list of all unique answers for the chosen

Simple Queries
(S-1) Who tested positive?
(S-2) How long did people wait for negative test results?
(S-3) Where were people not able to access testing?
(S-4) Who is promoting cures or preventions?
(S-5) Which companies have employers who tested positive?
Advanced Queries
(A-1) Who tested positive that had close contact with Boris Johnson?
(A-2) Who tested positive that traveled to Japan?
(A-3) What methods of cure and prevention do people think are effective?
(A-4) Who has died of COVID-19 that has a relationship with the author of the tweet?
(A-5) Who is showing symptoms but can not get tested?

Table 5.7: Queries used for evaluating our semantic search system. *Simple Queries* only involve GROUPBY operator. *Advanced Queries* contain both SELECT and GROUPBY operators.

slot, which are extracted from tweets that match the search criteria, sorted by mention frequency. For example, a user might enter *San Francisco* in the location field, and "$\star$" for EMPLOYER; the system will then return a list of all employers located in San Francisco where an employee was reported positive. We find this simple interface enables a rich set of informative queries over events that are automatically extracted by our models. A list of sample queries supported by TWICSS is presented in Table 5.7. Table 5.11 presents a sample of the outputs of our system.

As our search has to be done for tweets on the million level (see exact number of tweets in the following section), we deploy Elasticsearch in the back end to allow near real-time responses. Elasticsearch is a full-text search engine that relies on indices, where each document is associated with a collection of fields. In our case, we treat each tweet as a document and each slot as an associated field. All slots for each event type are indexed and thus searchable through search queries.

5.5.2 Evaluation

Precision of Top K Extractions. We evaluate the accuracy of answers returned by our semantic search system using 10 sample queries in Table 5.7. Table 5.11(a) presents the top 20 returned results from these queries. We then manually examine their top K extractions. Table 5.8 presents our results.[39] We observe that our system has high precision for nearly all queries, including queries involving slots with few annotations. For example, although DURATION is excluded in Table 5.5 because there are fewer than 20 instances in the test set, TWICSS still achieves good performance on some queries involving this slot, due to redundancy of information in Twitter.

No.	P@20	P@50	P@100
S-1	100	100	99
S-2	85	82	82
S-3	100	100	100
S-4	90	96	91
S-5	90	90	92

(a) General queries

No.	P@10	P@20	P@50
A-1	70	60	58
A-2	100	100	96
A-3	90	85	82
A-4	100	95	92
A-5	100	100	100

(b) Specific queries

Table 5.8: Precision@K of our semantic search system, using queries listed in Table 5.7.

[39]We treat an item to be a correct extraction if any of the associated tweets has correct corresponding filled slots.

Extracted Answer Types. We also analyze the types of answers that our system extracts, using queries that have GROUPBY operation towards the WHO slot. We define two answer types: (1) Specific entities normally have concrete meanings. It includes names for public figures, such as "Boris Johnson" and "Dominic Cummings"; (2) Generic entities are under-specified, typically nominal references, for example "my daughter" or "a woman". To understand how ordinary people post information related to their own situations, we also calculate the percentage of name entities that have relationship with the author of the tweet. Table 5.9 shows our results.

No.	# Correct	Specific (%)	Generic (%)	Personal (%)
S-1	99	63.6	36.4	7.0
S-4	91	75.8	24.2	2.0
A-1	29	100.0	0.0	0.0
A-2	48	0.0	100.0	0.0
A-4	46	100.0	0.0	92.0
A-5	50	4.0	96.0	92.0

Table 5.9: Breakdown analysis for the types of answers from our semantic search system. "Personal" refers to personal cases that are related to the author of the tweet.

5.5.3 Examples

In this section, we present search results for our example queries (in Table 5.11).

Tracking Self-Reported Cases. Our semantic search system can help track self-reported cases. For queries such as "Who is showing symptoms but can not get access to testing?", we can select out self-reported cases by restricting WHO slot to "I", "we" or "author of the tweet". From our semantic search system, we note 2,546 self-reported cases showing

symptoms but do not have access to testing, 903 of which are reported after 2020/07/01.[40]

Perceived Cure and Prevention Methods. We observe a variety of people or organizations are promoting cure or prevention methods for coronavirus, including "Dr. Fauci", "CDC" and "WHO". The top 5 cure or prevention methods that Twitter users believe effective are "social distancing", "hydroxychloroquine", "(wash) your hands", "masks" and "a mask" (shown in Table 5.11). We notice these methods are all among the top ranked items people think effective since January of 2020.

(A-1) Who tested positive that has close contact with Boris Johnson?		
Error Type	**Entity**	**Tweet**
Inference Error	jair bolsanaro	Jair Bolsanaro has tested positive for Covid-19. Noval Djokovic and Boris Johnson had it. Life sometimes comes a full circle very fast.
	the british pm	WH says Trump spoke with Boris Johnson and "wished him a speedy recovery" after the British PM tested positive for coronavirus.
Ambiguous Case	dominic cummings	Boris Johnson's senior adviser, Dominic Cummings, is self-isolating at home after developing #coronavirus symptoms. http://bbc.in/2WQhbsZ Last week, the PM and Health Secretary Matt Hancock both tested positive for #Covid19. WATCH: https://bbc.in/2Jv55xj #Newsnight

Table 5.10: Examples of errors made by our semantic search system.

5.5.4 Error Analysis

Based on our manual inspection, 72 incorrect extractions were identified in all 10 sample queries; these can be grouped into categories as follows: inference errors (58.3%), segmentation errors (23.6%), ambiguous cases (12.5%) and others (5.6%). We present examples of each error category in Table 5.10.

[40]To protect user privacy, we have hidden tweets which the tweet authors self-report the situations of themselves or their relatives.

Inference Errors. We notice our BERT based model struggles with slots that may involve subtle inferences (like CLOSE CONTACT or RELATION), although the limited number of annotations for these slots might also be a factor in this type of error. For example, in the first example in Table 5.10, the tweet does not imply that "Jair Bolsanaro" was in close contact with "Boris Johnson"; in the second example, the model fails to identify that "Boris Johnson" and "the British pm" refer to the same person.

Segmentation Errors. In some cases the extracted items contain extra tokens because of chunker errors, for example "georgia drank disinfectants" was extracted as a cure method. We also notice our choice of only extracting noun phrase chunks does not capture verb phrases for the CURE AND PREVENTION category. For example, instead of extracting "washing your hands" and "don't touch your face" as prevention methods, our system only extracts "your hands" and "your face".

Ambiguous Cases. In some cases, it is debatable whether an extraction is correct without additional context. For instance in the third example in Table 5.10, we don't know if "Dominic Cummings" tested positive, although the tweet seems to imply he might have been infected. We consider the extraction to be an error in this case, since the tweet didn't specifically mention that he tested positive.

5.6 Additional Analysis

Our annotations can be potentially used for other analytical purposes besides event slot filling. All following analyses are done on our train and dev set (7,500 tweets).

Demographics of Users. We are interested in the demographics of 348 unique users who self-report their situations. We hire an in-house annotator to manually annotate the demographics of users based on their public profiles. We observe the following interesting

trends: (1) The gender of self-reported users is roughly evenly split (female 55.7% versus male 44.3% for 271 users with gender information). (2) The top two categories for race are white (171 users) and black (34 users), excluding 92 users that we are not able to identify race. (3) Among 153 users whose political inclinations can be identified, 143 are democrats. (4) 55.6% of the 223 users who are reporting their ages are from age group 25-50. (5) 69.0% of the 348 users who self-reported information related to COVID-19 also disclose their geo-location information on Twitter. 74.2% of these users are located in the United States and 10.8% in European countries.

Bots and Organization Accounts. Within 7,326 unique users in our corpus, 2.4% are potentially bots, as identified by the Botometer API [116]. We also note 4.1% of tweets about CURE & PREVENTION are potentially posted by bots.[41] 18.5% of user accounts belong to organizations, according to the Humanizr [71].

5.7 Conclusion

In this chapter, we presented an annotated corpus of 10,000 tweets for COVID-19 events, including positive/negative tests and denied access to testing. We demonstrate that our corpus supports automatic extraction of answers for filling questions specific to each event. We further build a semantic search system over the extracted events that supports user-defined structured queries. We believe our semantic search system can be a useful tool to help address information overload for COVID-19 related information reported on Twitter.

[41] A user is considered as a bot if its Complete Automation Probability (CAP) from Botometer is ≥ 0.6.

(A-3) What are the cure methods that people think effective?			
social distancing (7,755)	hydroxychloroquine (5,156)	your hands (4,601)	masks (3,755)
a mask (3,339)	face masks (2,035)	physical distancing (1,593)	chloroquine (1,490)
home (1,079)	bleach (1,039)	your face (938)	a face mask (889)
a cloth face covering (823)	soap and water (782)	mask (775)	vitamin c (749)
eye protection (719)	widespread mask-wearing (705)	cannabis (688)	georgia drank disinfectants (643)

(a) Sample outputs with top 20 extracted answers (corresponding tweets are omitted). Numbers in brackets are frequency counts.

(A-3) What are the cure methods that people think effective?	
social distancing (count: 7,755)	The goal of social distancing was never to prevent COVID-19 from spreading completely; it was to prevent the virus from spreading so rapidly as to overwhelm our healthcare system. That goal has been largely achieved.
	Very good indeed but you need also to remind them keeping social distancing, another basic protective measure to prevent the spread of #covid19.
hydroxy-chloroquine (count: 5,156)	Our experience suggests that hydroxychloroquine should be a first-line treatment for Covid-19. We can use it to save lives and prevent others from becoming infected, write @DrJeffColyer and Daniel Hinthorn via @WSJ
	Hydroxychloroquine is now an official treatment for covid 19. Cheap cure with high efficacy & safe A good news !!
(A-5) Who is showing symptoms but can not get tested?	
My daughter (count: 862)	My daughter and I both symptomatic for over 3 weeks. Can't get tested. Very frustrating. She is in isolation and we don't know whether it is necessary or not. Psychological implications of not knowing dangerous for people with anxiety and depression.
	My daughter lives in NJ, is high risk, has all the symptoms, but can't get tested because she hasn't been out of the country and doesn't know which of her neighbors in the local hotspot have been positive.
My son (count: 684)	And you are in quarantine!! My son is very ill. Gone back to the DR 3 times, on 5 scripts. Can't get tested in your state. Difference between genius and stupidity is genius has it's limits.
	My son has symptoms and can't get a test. His doctor made very clear that no testing was available unless you've contacted a person with a known positive. Without tests no one is a known positive! This is on you.

(b) Sample outputs with top 2 extracted answers and 2 randomly sampled corresponding tweets.

Table 5.11: Sample outputs from our semantic search system.

Chapter 6: Conclusions and Future Work

In this thesis, we explore the event extraction task from large and heterogeneous user-generated informal text, such as Twitter and Web. Three different types of events (cybersecurity events, people's geopolitical and financial forecasting events, and COVID-19 events) are explored. Considering the different nature of text sources and our study purposes, we develop different methods for extracting events. For cybersecurity events (in Chapter 3) and COVID-19 events (in Chapter 5), we first use a set of carefully chosen keywords to collect tweets from Twitter. We then train machine learning models to automatically extract events, by using linguistic annotated corpus that we develop. All annotated datasets are available online. For people's financial forecasts (in Chapter 4), we make use of a rule-based approach to extract analysts' numerical forecast values. We find this approach to be of having both high precision and recall.

Besides extracting associated attributes for these events, we also focus on analyzing the extracted events for social goods. We demonstrate that extracted events could enable a variety of real-world applications that could benefit people. For example, in Chapter 3, we develop a system that could track cybersecurity threats reported on Twitter along with predicted severity scores; in Chapter 5, we develop a semantic search application based on around 5 million tweets that support user self-defined queries to search for COVID-19 related information. We also demonstrate that the extracted signals from text can be used

for understanding people's behaviors. For example, we perform the first linguistic analysis towards people's decision making process in Chapter 4.

6.1 Future Work

In this section, we outline extensions and future directions of our current work.

6.1.1 Overconfidence

As discussed in Chapter 1, there is a limited number of analyses for people's decision-making processes from a linguistic perspective. However, human makes different kinds of mistakes or errors when thinking and judging. People's overconfidence is one of these cognition illusions. The overconfidence phenomenon describes a miscalibration between people's subjective confidence and objective accuracy. It has been widely studied in psychology and social science community. There are mainly three types of overconfidence: (1) overestimation of one's actual performance, (2) overplacement of one's performance relative to others, and (3) overprecision of one's beliefs in an analysis [78]. For this part, we will mainly focus on the first type.

To the best of our knowledge, rare work has been done on identifying the linguistic cues for (over)confidence in computational linguistic community. [45] makes the first attempt of evaluating confidence and competence in an online group discussion setting. But they only measure people's confidence (rather than overconfidence), and formulate it as a binary classification problem (less confidence or more confident).

Thus, one future direction is to build a computational tool for testing the existence (and the degree) of overconfidence, both from personal level and individual forecast level. Specifically, we hope to evaluate the following hypotheses/questions: (1) Whether text is an indicator for identifying people's overconfidence? (2) More importantly, whether we could

quantify people's overconfidence level? Good Judgment Open dataset can be a potential dataset to build this computational model, as it explicitly links people's cognitive reasoning process with text. It also provides one way to compare people's subjective inceptions with an objective metric, as we know the final outcome of the forecasting events.

Bibliography

[1] Muhammad Abdul-Mageed and Lyle Ungar. EmoNet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 718–728, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[2] Heike Adel and Hinrich Schütze. Exploring different dimensions of attention for uncertainty detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 22–34, Valencia, Spain, April 2017. Association for Computational Linguistics.

[3] Luca Allodi and Fabio Massacci. A preliminary analysis of vulnerability scores for attacks in wild: The ekits and sym datasets. In *Proceedings of the 2012 ACM Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, BADGERS '12, pages 17–24, New York, NY, USA, 2012. ACM.

[4] Tim Althoff, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. How to ask for a favor: A case study on the success of altruistic requests, 2014.

[5] Sinan Aral, Lev Muchnik, and Arun Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, 106(51):21544–21549, 2009.

[6] Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008.

[7] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[8] David Bamman, Jacob Eisenstein, and Tyler Schnoebelen. Gender identity and lexical variation in social media. *Journal of Sociolinguistics*, 2014.

[9] Juan M. Banda, Ramya Tekumalla, Guanyu Wang, Jingyuan Yu, Tuo Liu, Yuning Ding, and Gerardo Chowell. A large-scale covid-19 twitter chatter dataset for open scientific research – an international collaboration, 2020.

[10] Russell M. Barefield and Eugene E. Comiskey. The accuracy of analysts' forecasts of earnings per share. *Journal of Business Research*, 3(3):241–252, 1975.

[11] Edward Benson, Aria Haghighi, and Regina Barzilay. Event discovery in social media feeds. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*, 2011.

[12] Adrian Benton, Margaret Mitchell, and Dirk Hovy. Multitask learning for mental health conditions with limited social media data. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017.

[13] Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1510, Doha, Qatar, October 2014. Association for Computational Linguistics.

[14] Mehran Bozorgi, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Beyond heuristics: Learning to classify vulnerabilities and predict exploits. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 105–114, New York, NY, USA, 2010. ACM.

[15] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 1950.

[16] Benjamin L. Bullough, Anna K. Yanchenko, Christopher L. Smith, and Joseph R. Zipkin. Predicting exploitation of disclosed software vulnerabilities using open-source data. In *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics*, IWSPA '17, pages 45–53, New York, NY, USA, 2017. ACM.

[17] Andrew C. Call, Shuping Chen, and Yen H. Tong. Are analysts' earnings forecasts more accurate when accompanied by cash flow forecasts? *Review of Accounting Studies*, 14(2):358–391, Sep 2009.

[18] Nathanael Chambers, Ben Fry, and James McMasters. Detecting denial-of-service attacks from social media text: Applying nlp to computer security. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1626–1635, 2018.

[19] Ching-Yun Chang, Zhiyang Teng, and Yue Zhang. Expectation-regulated neural model for event mention extraction. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 400–410, 2016.

[20] Emily Chen, Kristina Lerman, and Emilio Ferrara. Tracking social media discourse about the covid-19 pandemic: Development of a public coronavirus twitter data set. *JMIR Public Health Surveill*, 6(2):e19273, May 2020.

[21] Vijay Kumar Chopra. Why so much error in analysts' earnings forecasts? *Financial Analysts Journal*, 54(6):35–42, 1998.

[22] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[23] Glen Coppersmith, Mark Dredze, Craig Harman, and Kristy Hollingshead. From ADHD to SAD: Analyzing the language of mental health on twitter through self-reported diagnoses. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 1–10, Denver, Colorado, June 5 2015. Association for Computational Linguistics.

[24] Timothy Crichfield, Thomas Dyckman, and Josef Lakonishok. An evaluation of security analysts' forecasts. *The Accounting Review*, 53(3):651–668, 1978.

[25] Edgar Dale and Jeanne S. Chall. A formula for predicting readability. *Educational Research Bulletin*, 27(1):11–28, 1948.

[26] Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 250–259, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[27] Debopam Das, Tatjana Scheffler, Peter Bourgonje, and Manfred Stede. Constructing a lexicon of English discourse connectives. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 360–365, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[28] Munmun De Choudhury, Scott Counts, Eric J. Horvitz, and Aaron Hoff. Characterizing and predicting postpartum depression from shared facebook data. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '14, 2014.

[29] Marie-Catherine de Marneffe, Christopher D Manning, and Christopher Potts. Did it happen? the pragmatic complexity of veridicality assessment. *Computational linguistics*, 2012.

[30] Dorottya Demszky, Nikhil Garg, Rob Voigt, James Zou, Matthew Gentzkow, Jesse Shapiro, and Dan Jurafsky. Analyzing polarization in social media: Method and application to tweets on 21 mass shootings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[31] Dorottya Demszky, Nikhil Garg, Rob Voigt, James Zou, Jesse Shapiro, Matthew Gentzkow, and Dan Jurafsky. Analyzing polarization in social media: Method and application to tweets on 21 mass shootings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2970–3005, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[32] Shrey Desai, Cornelia Caragea, and Junyi Jessy Li. Detecting perceived emotions in hurricane disasters. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5290–5305, Online, July 2020. Association for Computational Linguistics.

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[34] Dimitar Dimitrov, Erdal Baran, Pavlos Fafalios, Ran Yu, Xiaofei Zhu, Matthäus Zloch, and Stefan Dietze. Tweetscov19 – a knowledge base of semantically annotated tweets about the covid-19 pandemic, 2020.

[35] Doug Downey, Oren Etzioni, and Stephen Soderland. A probabilistic model of redundancy in information extraction. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, page 1034–1041, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

[36] Mark Dredze, Miles Osborne, and Prabhanjan Kambadur. Geolocation for twitter: Timing matters. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, pages 1064–1069, San Diego, California, June 2016. Association for Computational Linguistics.

[37] David N. Dreman and Michael A. Berry. Analyst forecasting errors and their implications for security analysis. *Financial Analysts Journal*, 51(3):30–41, 1995.

[38] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[39] Johannes C Eichstaedt, Hansen Andrew Schwartz, Margaret L Kern, Gregory Park, Darwin R Labarthe, Raina M Merchant, Sneha Jha, Megha Agrawal, Lukasz A Dziurzynski, Maarten Sap, et al. Psychological language on twitter predicts county-level heart disease mortality. *Psychological science*, 2015.

[40] Johannes C. Eichstaedt, Robert J. Smith, Raina M. Merchant, Lyle H. Ungar, Patrick Crutchley, Daniel Preoţiuc-Pietro, David A. Asch, and H. Andrew Schwartz. Facebook language predicts depression in medical records. *Proceedings of the National Academy of Sciences*, 115(44):11203–11208, 2018.

[41] Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1277–1287, Cambridge, MA, October 2010. Association for Computational Linguistics.

[42] Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. The CoNLL-2010 shared task: Learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 1–12, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[43] Rudolph Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233, 1948.

[44] Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. Large scale crowdsourcing and characterization of twitter abusive behavior, 2018.

[45] Liye Fu, Lillian Lee, and Cristian Danescu-Niculescu-Mizil. When confidence and competence collide: Effects on online decision-making discussions. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1381–1390, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[46] Vikas Ganjigunte Ashok, Song Feng, and Yejin Choi. Success with style: Using writing style to predict the success of novels. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1764, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[47] Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, John Barnden, and Antonio Reyes. SemEval-2015 task 11: Sentiment analysis of figurative language in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 470–478, Denver, Colorado, June 2015. Association for Computational Linguistics.

[48] Jon Gillick and David Bamman. Please clap: Modeling applause in campaign speeches. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

[49] Jeremy Ginsberg, Matthew H Mohebbi, Rajan S Patel, Lynnette Brammer, Mark S Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, February 2009.

[50] Bo Han and Timothy Baldwin. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[51] Bo Han, Afshin Rahimi, Leon Derczynski, and Timothy Baldwin. Twitter geolocation prediction shared task of the 2016 workshop on noisy user-generated text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 213–217, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[52] Zhuobing Han, Xiaohong Li, Zhenchang Xing, Hongtao Liu, and Zhiyong Feng. Learning to predict severity of software vulnerability using only vulnerability description. In *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 125–136, Sept 2017.

[53] Tamanna Hossain, Robert L. Logan IV, Arjuna Ugarte, Yoshitomo Matsubara, Sean Young, and Sameer Singh. Detecting covid-19 misinformation on social media, 2020.

[54] Dirk Hovy, Anders Johannsen, and Anders Søgaard. User review sites as a resource for large-scale sociolinguistic studies. In *Proceedings of the 24th international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015.

[55] Heng Ji, Ralph Grishman, and Hoa Dang. Overview of the TAC2011 knowledge base population track. In *TAC 2011 Proceedings Papers*, 2011.

[56] Kristen Johnson, Di Jin, and Dan Goldwasser. Leveraging behavioral and social information for weakly supervised collective classification of political discourse on twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 741–752, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[57] Daniel Jurafsky and James H Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, 2000.

[58] Katherine Keith and Amanda Stent. Modeling financial analysts' decision making via the pragmatics and semantics of earnings calls. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 493–503, Florence, Italy, July 2019. Association for Computational Linguistics.

[59] Emre Kiciman, Scott Counts, and Melissa Gasser. Using longitudinal social media analysis to understand the effects of early college alcohol use. June 2018.

[60] Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1–9, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[61] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[62] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[63] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[64] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[65] Shimon Kogan, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American*

Chapter of the Association for Computational Linguistics, pages 272–280, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[66] Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[67] David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-László Barabási, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, Tony Jebara, Gary King, Michael Macy, Deb Roy, and Marshall Van Alstyne. Computational social science. *Science*, 323(5915):721–723, 2009.

[68] Roger K. Loh and G. Mujtaba Mian. Do accurate earnings forecasts facilitate superior investment recommendations? *Journal of Financial Economics*, 80(2):455 – 483, 2006.

[69] Tim Loughran and Bill Mcdonald. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65, 2011.

[70] Marco Lui and Timothy Baldwin. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[71] James McCorriston, David Jurgens, and Derek Ruths. Organizations are users too: Characterizing and detecting the presence of organizations on twitter. In *Proceedings of the 9th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2015.

[72] Barbara Mellers, Eric Stone, Pavel Atanasov, Nick Rohrbaugh, S Emlen Metz, Lyle Ungar, Michael M Bishop, Michael Horowitz, Ed Merkle, and Philip Tetlock. The psychology of intelligence analysis: Drivers of prediction accuracy in world politics. *Journal of Experimental Psychology: Applied*, 21, 2015.

[73] Barbara Mellers, Eric Stone, Terry Murray, Angela Minster, Nick Rohrbaugh, Michael Bishop, Eva Chen, Joshua Baker, Yuan Hou, Michael Horowitz, et al. Identifying and cultivating superforecasters as a method of improving probabilistic predictions. *Perspectives on Psychological Science*, 2015.

[74] Barbara Mellers, Lyle Ungar, Jonathan Baron, Jaime Ramos, Burcu Gurcay, Katrina Fincher, Sydney E. Scott, Don Moore, Pavel Atanasov, Samuel A. Swift, Terry Murray, Eric Stone, and Philip E. Tetlock. Psychological strategies for winning a geopolitical forecasting tournament. *Psychological Science*, 25(5):1106–1115, 2014. PMID: 24659192.

[75] Bonan Min and Ralph Grishman. Compensating for annotation errors in training a relation extractor. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 194–203, 2012.

[76] Saif Mohammad. #emotional tweets. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 246–255, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics.

[77] Saif M. Mohammad and Peter D. Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[78] Don A Moore and Paul J Healy. The trouble with overconfidence. *Psychological Review*, 115(2):502–517, 2008.

[79] Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. SemEval-2013 task 2: Sentiment analysis in twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[80] Dat Quoc Nguyen, Thanh Vu, Afshin Rahimi, Mai Hoang Dao, Linh The Nguyen, and Long Doan. WNUT-2020 Task 2: Identification of Informative COVID-19 English Tweets. In *Proceedings of the 6th Workshop on Noisy User-generated Text*, 2020.

[81] Dong Nguyen, Rilana Gravel, Dolf Trieschnigg, and Theo Meder. " how old do you think i am?" a study of language and age in twitter. In *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[82] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, 2010.

[83] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002.

[84] Gregory Park, H Andrew Schwartz, Maarten Sap, Margaret L Kern, Evan Weingarten, Johannes C Eichstaedt, Jonah Berger, David J Stillwell, Michal Kosinski,

Lyle H Ungar, et al. Living in the past, present, and future: Measuring temporal orientation with language. *Journal of personality*, 2017.

[85] Michael J Paul, Mark Dredze, and David Broniatowski. Twitter improves influenza forecasting. *PLOS Currents Outbreaks*, 2014.

[86] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[87] Verónica Pérez-Rosas and Rada Mihalcea. Experiments in open domain deception detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[88] Umair Qazi, Muhammad Imran, and Ferda Ofli. Geocov19: A dataset of hundreds of millions of multilingual covid-19 tweets with location information. *SIGSPATIAL Special*, 12(1):6–15, June 2020.

[89] Jishnu Ray Chowdhury, Cornelia Caragea, and Doina Caragea. Keyphrase extraction from disaster-related tweets. In *The World Wide Web Conference*, WWW '19, page 1555–1566, New York, NY, USA, 2019. Association for Computing Machinery.

[90] Alan Ritter, Colin Cherry, and Bill Dolan. Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180, Los Angeles, California, June 2010. Association for Computational Linguistics.

[91] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[92] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[93] Alan Ritter, Oren Etzioni, Sam Clark, et al. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

[94] Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, page 1104–1112, New York, NY, USA, 2012. Association for Computing Machinery.

[95] Alan Ritter, Evan Wright, William Casey, and Tom Mitchell. Weakly supervised extraction of computer security events from twitter. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 896–905, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee.

[96] Alan Ritter, Evan Wright, William Casey, and Tom Mitchell. Weakly supervised extraction of computer security events from twitter. In *Proceedings of the 24th International Conference on World Wide Web*, pages 896–905. International World Wide Web Conferences Steering Committee, 2015.

[97] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, 2015.

[98] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 1041–1056, Washington, D.C., 2015. USENIX Association.

[99] Roser Saurí and James Pustejovsky. Are you sure that this happened? assessing the factuality degree of events in text. *Computational Linguistics*, 2012.

[100] Alan G. Sawyer, Juliano Laran, and Jun Xu. The readability of marketing journals: Are award-winning articles better written? *Journal of Marketing*, 72(1):108–117, 2008.

[101] H Andrew Schwartz, Gregory Park, Maarten Sap, Evan Weingarten, Johannes Eichstaedt, Margaret Kern, David Stillwell, Michal Kosinski, Jonah Berger, Martin Seligman, et al. Extracting human temporal orientation from facebook language. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015.

[102] H. Andrew Schwartz, Masoud Rouhizadeh, Michael Bishop, Philip Tetlock, Barbara Mellers, and Lyle Ungar. Assessing objective recommendation quality through political forecasting. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2348–2357, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[103] Noah A. Smith. *Text-driven forecasting*. 2010.

[104] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[105] Thamar Solorio, Ragib Hasan, and Mainul Mizan. A case study of sockpuppet detection in wikipedia. In *Proceedings of the Workshop on Language Analysis in Social Media*, 2013.

[106] Youngseo Son, Nipun Bayas, and H Andrew Schwartz. Causal explanation analysis on social media. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[107] Sandesh Swamy, Alan Ritter, and Marie-Catherine de Marneffe. " i have a feeling trump will win.....": Forecasting winners and losers from user predictions on twitter. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[108] Sandesh Swamy, Alan Ritter, and Marie-Catherine de Marneffe. "i have a feeling trump will win.....": Forecasting winners and losers from user predictions on twitter. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1583–1592, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[109] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307, June 2011.

[110] Chenhao Tan, Lillian Lee, and Bo Pang. The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 175–185, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[111] Chenhao Tan, Lillian Lee, and Bo Pang. The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. In *ACL*, 2014.

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

[113] Yla R. Tausczik and James W. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 2010.

[114] Philip Tetlock. Expert political judgment: How good is it? how can we know? 2005.

[115] Oren Tsur and Ari Rappoport. Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews. In *Third International AAAI Conference on Weblogs and Social Media*, 2009.

[116] Onur Varol, Emilio Ferrara, Clayton Davis, Filippo Menczer, and Alessandro Flammini. Online human-bot interactions: Detection, estimation, and characterization. In *Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM)*, 2017.

[117] Rob Voigt, Nicholas P. Camp, Vinodkumar Prabhakaran, William L. Hamilton, Rebecca C. Hetey, Camilla M. Griffiths, David Jurgens, Dan Jurafsky, and Jennifer L. Eberhardt. Language from police body camera footage shows racial disparities in officer respect. *Proceedings of the National Academy of Sciences*, 114(25):6521–6526, 2017.

[118] Rob Voigt, Nicholas P Camp, Vinodkumar Prabhakaran, William L Hamilton, Rebecca C Hetey, Camilla M Griffiths, David Jurgens, Dan Jurafsky, and Jennifer L Eberhardt. Language from police body camera footage shows racial disparities in officer respect. *Proceedings of the National Academy of Sciences*, 2017.

[119] Rob Voigt, David Jurgens, Vinodkumar Prabhakaran, Dan Jurafsky, and Yulia Tsvetkov. Rtgender: A corpus for studying differential responses to gender. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

[120] Svitlana Volkova, Glen Coppersmith, and Benjamin Van Durme. Inferring user political preferences from streaming communications. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014.

[121] Zijian Wang and David Jurgens. It's going to be okay: Measuring access to support in online communities. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 33–45, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[122] Wei Wei, Kenneth Joseph, Wei Lo, and Kathleen M Carley. A bayesian graphical model to discover latent events from twitter. In *ICWSM*, 2015.

[123] Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. Learning subjective language. *Computational linguistics*, 2004.

[124] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[125] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005.

[126] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[127] Diyi Yang, Jiaao Chen, Zichao Yang, Dan Jurafsky, and Eduard Hovy. Let's make your request more persuasive: Modeling persuasive strategies via semi-supervised neural nets on crowdfunding platforms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[128] Justine Zhang, Robert Filbin, Christine Morrison, Jaclyn Weiser, and Cristian Danescu-Niculescu-Mizil. Finding your voice: The linguistic development of mental health counselors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 936–947, Florence, Italy, July 2019. Association for Computational Linguistics.

[129] Deyu Zhou, Liangyu Chen, and Yulan He. A simple bayesian modelling approach to event extraction from twitter. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014.

[130] Deyu Zhou, Xuan Zhang, and Yulan He. Event extraction from Twitter using non-parametric Bayesian mixture model with word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 808–817, Valencia, Spain, April 2017. Association for Computational Linguistics.

[131] Shi Zong, Alan Ritter, Graham Mueller, and Evan Wright. Analyzing the perceived severity of cybersecurity threats reported on social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 1380–1390, 2019.