Lambretta

Learning to Rank for Twitter Soft Moderation

Paudel, Pujan; Blackburn, Jeremy; De Cristofaro, Emiliano; Zannettou, Savvas; Stringhini, Gianluca

DOI
10.1109/SP46215.2023.10179392

Publication date
2023

Document Version
Final published version

Published in
Proceedings - 44th IEEE Symposium on Security and Privacy, SP 2023

LAMBRETTA: Learning to Rank for Twitter Soft Moderation

Pujan Paudel[♣], Jeremy Blackburn[♦], Emiliano De Cristofaro[♥], Savvas Zannettou[♠], and Gianluca Stringhini[♣]
[♣]Boston University, [♦]Binghamton University, [♥]University College London, [♠]Delft University of Technology
{ppaudel,gian}@bu.edu, jblackbu@binghamton.edu, e.decristofaro@ucl.ac.uk, s.zannettou@tudelft.nl

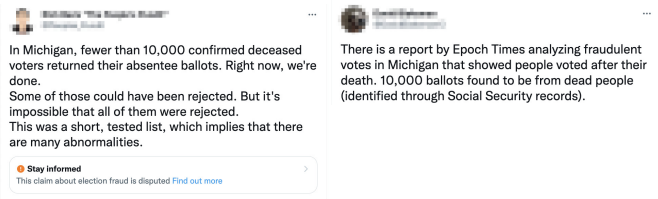
2023 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-9336-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/SP46215.2023.10179392

***Abstract*—To curb the problem of false information, social media platforms like Twitter started adding warning labels to content discussing debunked narratives, with the goal of providing more context to their audiences. Unfortunately, these labels are not applied uniformly and leave large amounts of false content unmoderated. This paper presents LAMBRETTA, a system that automatically identifies tweets that are candidates for soft moderation using Learning To Rank (LTR). We run LAMBRETTA on Twitter data to moderate false claims related to the 2020 US Election and find that it flags over 20 times more tweets than Twitter, with only 3.93% false positives and 18.81% false negatives, outperforming alternative state-of-the-art methods based on keyword extraction and semantic search. Overall, LAMBRETTA assists human moderators in identifying and flagging false information on social media.**

I. INTRODUCTION

The security research community has consistently been at the forefront of the fight against online abuse, from spam [27, 47], phishing [32, 108], to online fraud [15, 68, 94]. Today, one of the most pressing types of abuse is the spread of false information, especially on social networks. Arguably, mitigating it faces some unique challenges. First, while some malicious actors spread misleading/false claims to advance their goals (*"disinformation"*), false narratives are often believed by real users in good faith, who then re-share them on social media (*"misinformation"*) [42, 81, 85, 89, 93, 102, 105]. Second, identifying what is true or false is challenging, hard to automate, and often depends on external fact-checkers. Finally, online platforms are often concerned about the effects of taking action on dis- and misinformation; for example, limiting what is allowed to be said on a platform can raise concerns about censorship and reduce engagement (and thus profit) [33, 39, 61]. Nevertheless, the computer security research community is well poised to develop effective mitigation strategies for the problem of false online information, as highlighted by recent research in top tier venues in the field [58, 76, 81].

As part of their mitigation strategy, social networks have begun to adopt so-called *soft moderation*. Rather than removing content or banning accounts, they notify other users about false narratives and provide additional context. Warning labels are attached to posts containing potentially false, misleading, or harmful claims, e.g., in the context of political disinformation [104] or COVID-19 [49, 86]. An estimated 300,000 tweets were labeled under Twitter's Civic Integrity Policy [101] as misleading around the 2020 US Elections, accounting for 0.2% of all election-related tweets posted during that period. Twitter reported that warning labels applied to tweets in late 2021 resulted in noticeable decreases in replies, retweets, and likes (13%, 10%, and 15% reduction, respectively) [101].



(a) Moderated **(b)** Unmoderated

Figure 1: Two example tweets discussing alleged voting fraud in the State of Michigan during the 2020 US Presidential Election. Twitter added a warning label to the first but not the second one.

Motivation. Early soft moderation results have been encouraging. Prior work shows that warning labels may prompt site users to debunk false claims [104] or that they may reduce user interactions and extremism in comments [67]. Unfortunately, details of Twitter's methodology are not publicly known. Worse yet, recent work indicates that soft moderation may not be applied uniformly by Twitter, with "benign" tweets being erroneously labeled while misleading content goes unlabeled [7, 104]. For example, in Figure 1, we show two tweets discussing the same debunked narrative that fraudulent votes were cast for 10,000 deceased individuals in Michigan during the 2020 US Presidential election; one received the warning label, and one did not.

This example highlights the need for effective automated approaches to flag potentially misleading posts on social media. Such approaches should cover as many misleading posts as possible while minimizing the number of unrelated posts that receive soft moderation labels to avoid *alert fatigue* effects, where users start ignoring warnings if they become too frequent [49]. Recent work found that using overly-broad rules when applying soft-moderation labels (e.g., the inclusion of a specific hashtag) flags a large amount of unrelated content: 37% of TikTok videos received COVID-19 related soft moderation were false positives [49].

Research Objectives. In this paper, we set out to develop an automated system to flag candidate Twitter posts for soft moderation. To minimize the false positive problem observed in previous work, instead of adopting a *topic-specific* mod-

DOI 10.1109/SP46215.2023.00088

eration approach (e.g., moderating any tweet containing a specific keyword), we follow a *claim-specific* methodology, where moderation labels directly address the statement or claim made in the content they are applied to [60]. This fine-grained approach allows platforms to tailor warnings for specific claims, adjust severity cues on warning labels, and increase transparency by providing *explainable* labels on why a specific tweet was moderated.

Technical Roadmap. We introduce LAMBRETTA, a system to assist in detecting candidate posts for soft moderation using a claim-specific approach. LAMBRETTA takes as seed input a list of tweets moderated by Twitter and uses a Learning To Rank (LTR) based method [11] to extract the optimal set of keywords that characterize the posts related to the same tweets. E.g., from tweets in Figure 1, LAMBRETTA would extract *"michigan,dead,ballot"* as the best set of keywords to characterize discussion surrounding this claim in *both* tweets. LAMBRETTA then uses these keywords to find more candidates for soft moderation. We instantiate LAMBRETTA using the publicly available set of 2,224 tweets soft moderated by Twitter during the 2020 US Presidential Election, as curated by [104].[1] LAMBRETTA retrieves the best set of keywords from 900 claims extracted from those tweets, identifying 2,042,173 additional candidates for soft moderation.

Main Contributions & Findings. We show that LAMBRETTA performs better than alternative approaches from keyword extraction and semantic search when recommending candidate tweets for moderation. By manually analyzing the tweets flagged by our system, we find that our results are accurate, with 3.93% false positives and 18.81% false negatives, with both metrics being substantially lower than those reported by other approaches. Moreover, LAMBRETTA reduces the number of tweets a human moderator would have to review by over five times compared to the second-best algorithm. We also find that LAMBRETTA flags 20 times more candidates for moderation than those moderated by Twitter, suggesting that our approach can complement existing systems and improve the state of content moderation.

LAMBRETTA is platform-independent and can be bootstrapped from a single post corresponding to a narrative or an event deemed intervention-worthy by human moderators. With the ability to scale to thousands of posts from a single misleading post, we show that event-driven keyword detection systems can be used as a foundation for large-scale soft moderation intervention systems. At the same time, our results show that moderating content is a nuanced problem. For example, posts discussing false narratives often attempt to debunk or ridicule them as satire. Thus, we see LAMBRETTA as a tool to *help* moderators identify potential candidates for soft moderation while leaving the final decision to humans. We make LAMBRETTA's source code and the labeled dataset used in this paper publicly available.[2]

[1] https://github.com/zsavvas/Soft-Moderation-Interventions-Twitter

[2] https://github.com/idramalab/lambretta

II. DATASETS

Our experiments use three datasets. One includes the tweets that were soft-moderated by Twitter and is used as ground truth; the other two allow LAMBRETTA to find more tweets that are candidates for moderation in the wild.

Ground truth dataset. Our first dataset, denoted as $\mathbf{G}_1$, consists of the tweets that received soft moderation by Twitter released by Zannettou [104]. It contains 2,244 tweets posted by 853 users.[3]

Evaluation datasets. Two more datasets are used by LAMBRETTA to compile a set of tweets that are candidates for moderation. One, denoted as $\mathbf{D}_1$, is released by Abilov et al. [1]; the other, denoted as $\mathbf{D}_2$, is obtained by querying Twitter's Academic Research full-archive search endpoint. $\mathbf{D}_1$ contains 7.6M tweets related to the 2020 US Election. More precisely, the authors of [1] retrieve tweets from the Streaming API using a set of hashtags related to the voter fraud narrative surrounding the 2020 US Election (e.g., #ballotfraud, #voterfraud, #electionfraud, #stopthesteal). The dataset is available as a list of tweet IDs. After retrieving the complete tweet information from the Twitter API using these IDs we obtain 4,017,259 tweets. Unlike $\mathbf{D}_1$, $\mathbf{D}_2$ is built at runtime leveraging Twitter's Academic Research full-archive search endpoint. At various stages, we query this endpoint using keyword-based search, for example, as part of deriving features for the underlying ranking model. The two different datasets capture two different types of data availability scenarios, and allow us to test LAMBRETTA on the entire Twitter archive using $\mathbf{D}_2$. Note that to match the time frame of our ground truth $\mathbf{G}_1$, we only extract tweets for the period between November 1 and December 31, 2020. In the rest of the paper, we refer to the evaluation datasets $\mathbf{D}_1$ and $\mathbf{D}_2$ as "data store."

Ethical Considerations. Analyzing large-scale social media data may raise ethical concerns. In this paper, we only collect public Twitter data using the official API and do not interact with users. As such, this work is not considered human subjects research by our institution. Regardless, we are mindful of the privacy of Twitter users and do not analyze any personally identifiable information (e.g., location data, account names, etc.). Also, when presenting example tweets in this paper, we apply "heavy disguise" and paraphrase them to make user re-identification more difficult [20].

III. OVERVIEW OF LAMBRETTA

This section presents the different components of our system and how the end-to-end pipeline operates. LAMBRETTA takes a seed list of tweets discussing a false claim and identifies similar candidate tweets for moderation. A high-level overview of our system is presented in Figure 2. LAMBRETTA has three stages: 1) extracting claim structures from tweets, 2) training a Learning to Rank (LTR) model from claim structures to extract the most relevant set of keywords for a given claim,

[3] Note that the dataset also contains 16,571 quoted tweets with warning labels; we exclude them since they are mostly used as additional commentary on the original tweet itself.

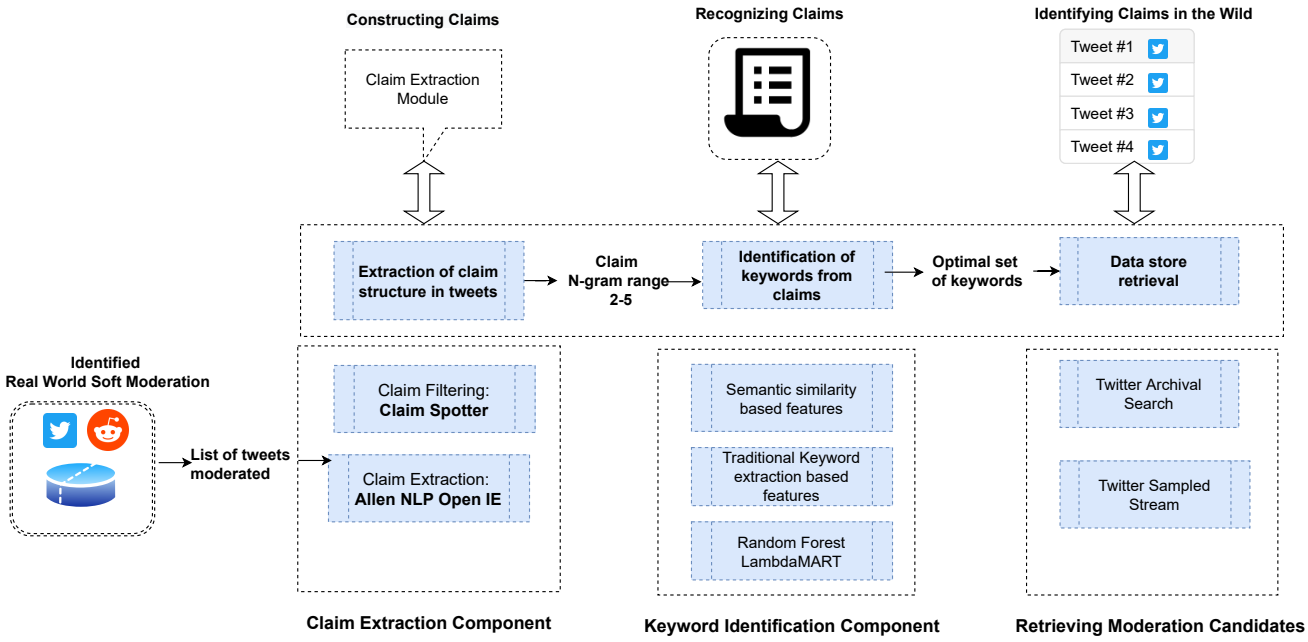


Figure 2: System overview of LAMBRETTA.

3) identifying candidate tweets from the data store similar to the seed tweets for platform moderators to make further decisions on.

A. *Claim structure extraction*

LAMBRETTA extracts claims that should be moderated, starting from one or more tweets. These extracted claims have similar structures to those manually curated by fact-checking organizations like Snopes[4] or PolitiFact.[5] Automatically extracting claims from tweets is useful for our purposes for two reasons. First, it narrows down spans of tweets containing assertions and provides a better training dataset for LAMBRETTA's subsequent phases. Second, while we could directly rely on the claims published by fact checkers instead of building them ourselves, this is not ideal since fact checks tend to lag behind the appearance of false narratives by 10-20 hours [84], which impairs real-time moderation.

To identify claims in the seed list of tweets, we follow an approach inspired by linguistics. First, we identify propositions in tweets, defined as "a declarative piece of text used to make a statement or assertion [65]." However, not all linguistic propositions contain a claim that should be part of soft moderation efforts. To account for this, LAMBRETTA runs identified propositions through a claim classification component. The output of this phase is a set of claims that can be passed to LAMBRETTA's later stages.

Extracting propositions. As an operational definition, we consider a proposition as a standalone span of the tweet, potentially containing a claim [65]. Not all propositions are a claim, despite being a statement or assertion. We will discuss the detailed specifications of what constitutes a claim later.

It is important to note that one tweet might contain multiple sentences, and a single sentence might contain multiple propositions. We draw motivations from prior work on claim extraction on Twitter [48], where the authors formulate the problem by using propositions extracted through Open Information Extraction (Open IE). The major advantage of using Open IE is the ability to extract an unbounded number of propositions in a sentence without requiring any domain knowledge of the underlying text, making it usable off-the-shelf. In a similar problem setting [48], an Open IE system called Clausie [17] efficiently clustered tweets related to two real-world events into twelve different claims. In our implementation of LAMBRETTA, we use a tool developed by Stanovsky et al. [92], which frames Open IE as a sequence tagging problem and uses a bi-LSTM transducer, available via Allen NLP [24]. For the rest of the paper, we refer to this Allen NLP Open IE extractor as the "Proposition Extractor." As an example, using the Proposition Extractor on a tweet with the body "Mail-in voting eases access barriers that might otherwise exclude voters physically unable to cast votes and has shown increased turnout across demographic groups." extracts the following sets of propositions: 1) "Mail-in voting intended to ease access barriers", 2) "access barriers that might otherwise exclude voters physically unable to cast votes," and 3) "Mail-in voting has shown increased turnout across demographic groups."

Building ground truth annotations. As a natural first step, we need to validate the applicability of the Proposition Extractor in extracting potential *claim* structures. We thus sample 200 random tweets from $\mathbf{G}_1$ and annotate spans that contain a claim. Before annotating spans in a tweet containing claims,

[4]https://www.snopes.com/fact-check/
[5]https://www.politifact.com/factchecks/

we need to formally define the scope of a claim. The lead author manually annotates our 200 tweet sample by adapting the scheme proposed in [40], which identified six different categories of claims via crowdsourced annotations of sentences extracted from subtitles of four UK political TV shows. We consider a tweet as containing a claim if the tweet:

- discusses an event occurring in a particular US state related to the 2020 elections;
- makes an assertion that can be further validated or debunked;
- cites events and stories related to election manipulation;
- references statistics to support or potentially deny an argument for election manipulation;
- reports breaking news or a developing event related to the 2020 election; or
- mentions actors often linked to conspiracy theories (Soros, Clinton, etc.) in relation to election events.

In the end, we find 115 tweets with at least a single claim in our random sample of 200.

To annotate claim *spans*, which are normalized units of claims embedded in a tweet, we follow the method outlined by Wuhrl and Klinger, who perform claim detection on biomedical tweets related to topics like COVID-19, measles, and depression [103]. They present a guideline to human annotators on the task of annotating claim spans within a tweet. From the guidelines, claim spans are the central statement of an argumentative structure, phrased with an argumentative intent while expressing a clear stance (support or oppose) about the argument.

This results in a ground truth dataset of 144 claim spans for the 115 tweets that contain a claim. To exemplify the type of claims we identified, we present the following three tweets, with the claim spans in bold.

- PA election night narrow lead was erased by **hundreds of thousands of mail-in ballots counted without Republican observer.** I wonder why were Republican observers excluded from the counting?
- Sidney Powell (part of Giuliani's team) is explaining that the **plan to steal the election from Trump was Hugo Chavez**. However, Chavez died in March of 2013.
- A September 2020 study released by Judicial Watch revealed that **U.S. counties had 1.8M more registered voters than eligible** voting-age citizens. That means that in these counties the **registration rate exceeded 100% of eligible voters.**

Some examples of tweets that do not contain a claim are:

- "Republicans need to rise together and stand against the fraud. Otherwise, our party might never win election again.'
- "Dems thought they would get away with cheating. But they got caught. Funny enough how quiet they are? I would fight really hard to clear my name if I was accused of cheating. Silence speaks louder than words."

We test the Proposition Extractor's ability to recognize all potential claim structures present in a tweet as propositions. First, we split the 200 tweets using a sentence tokenizer and query each sentence against the Proposition Extractor. We then compare the set of propositions returned by Proposition Extractor against the annotated ground truth spans, looking for missed ground truth spans. These types of errors are called *uninformative extractions* [19], where Open IE systems can omit critical information from the sentences in the propositions they identify. The Proposition Extractor extracts 264 propositions from 115 tweets containing claims, and 159 propositions from 85 tweets without claims. Of the 144 claim spans in our ground truth, the Proposition Extractor misses only four instances.

Filtering propositions that are claims. Thus far, we have 264 propositions extracted from 115 tweets, and only 144 of these propositions are annotated as claims. As discussed above, a tweet containing a claim span can have multiple other propositions that are not a claim, and we need to filter out these too. As a final step, LAMBRETTA needs to distinguish propositions that are claims from those that are not. We address this by further scoring the propositions extracted by the Proposition Extractor against a state-of-the-art claim classifier called ClaimSpotter [31], which leverages a gradient-based adversarial training on transformer networks to identify claims and is trained on manually labeled sentences from historical US presidential debates [55]. For any input text, ClaimSpotter returns a score between 0 and 1, representing how "check-worthy" a claim is. This is not how likely something is to be *true*, but instead if it makes an objective claim (1) vs. espousing a subjective opinion (0). For instance, a relatively subjective proposition such as *"And that, ladies and gentlemen, is how you steal an election"* returns a score of **0.285**, whereas a relatively objective proposition such as *"In the past 20 years there have been approx 250 million votes cast and around 1200 proven cases of voter fraud"* scores **0.85**.

While the ClaimSpotter API is already trained and publicly available, we need to devise an appropriate threshold that gives us accurate results for classification considering our problem setting. To this end, we obtain the Claim Spotter scores for all 144 claim spans annotated as ground truth and set the target class for these scores as 1. Similarly, we obtain the ClaimSpotter scores for the 159 propositions extracted from the 85 tweets without claims and the 120 propositions extracted from the 115 tweets with claims. We first perform a 75–25 train-test split to identify the optimal threshold of 0.490 based on the Receiver Operating Characteristic (ROC) curve. We then perform a 5-fold cross validatio- using this threshold, obtaining an average Precision of 0.881 and an average Recall of 0.895. The system misclassifies claim propositions as non-claim 3.46% of the time, while non-claim propositions are classified as claims with a rate of 6.58%. We deem these results acceptable and thus adopt 0.490 as our threshold for LAMBRETTA's claim extraction component.

Extracting claims from the entire dataset. From the remaining 2,044 tweets in $\mathbf{G}_1$ we extract 4,471 propositions, 756 of which are identified as claims based on our decision threshold. In addition to the 144 claims identified during the decision

threshold calibration phase, this brings us to 900 total claims extracted from 2,244 tweets in $\mathbf{G}_1$. Some examples of tweets that contain claims include:

- "The voting machine in Green Bay ran out of ink, which delayed the final results. An election official went back to City Hall to get more ink."
- "The Dominion Voting Systems machines switched over 6,000 ballots for Biden in Michigan. They are used throughout the United States on a wide scale in sixteen states. Virtually all of these states are blue states. The system needs to be audited to make sure the right person is elected."

These extracted claims are topically diverse, covering different aspects of the election process as multiple phases of the election event developed. We find many claims discussing mail-in ballots and ballot harvesting posed as a threat (e.g., *"poll workers stuffing ballot box with mail-in ballots for Democrats"*) and discussion of election fraud caused by voting machines (e.g., *"Dominion machine flipped votes from Trump to Biden"*). Additionally, we find many claims misinterpreting bipartisan events of ballot counting as partisan vote counting (e.g., *"spike for Biden votes as suitcases of ballots started to be scanned"*) and presence of dead, fake, and ineligible voters (e.g., *"86,845 mail-in ballots lost in Arizona"*). Finally, many claims in our dataset discuss the developing situation of the vote totals as statistically suspicious or rigged (e.g., *"mysterious spike of votes in Pennsylvania had 600,000 votes for Biden and only 3,200 for Trump"*). This diversity of topics found in the claims will be an important evaluation setup for LAMBRETTA and its ability to generalize over various topics about an event.

B. Identification of keywords from claims

After summarizing tweets into claims, LAMBRETTA extracts the most representative set of keywords from claims. We need to further extract keywords from our claims because querying Twitter search with the full claim text will give us a very limited set of tweets discussing the claim in a narrow way. The following are three tweets discussing the same claim related to voting machines:

Claim: "smartmatic foreign software voting machine design rig election socialist venezuela."
Example Tweets discussing the same claim:

- "Dominion developed by Sequoia-Smartmatic, a company previously owned by Chavez. US Intel said Smartmatic was used to rig the 2004 election in Venezuela. The Chicago election commission concluded the Smartmatic software delivers the results desired to the election officials."
- "China, Cuba Venezuela, all these communist countries and Antifa interfered in U.S elections. They used Smartmatic voting software, created by Hugo Chavez. Guess who helped them : George Soros and the Clinton Foundation, rigging U.S election for Joe Biden."
- "The Dominion Smartmatic machine was used in Venezuela by Cuban intelligence in an attack on its democracy by Chavez and Castro. Warfare through Voting systems has replaced guerrilla attacks against nations."

The first tweet has a very formal tone to it, is detailed, and uses some sources (US Intel, Chicago Election Commission) to establish context. On the other hand, the second tweet also talks about Smartmatic and its connection with Venezuela, but has totally different actors surrounding the discussion (Antifa, George Soros, and the Clinton Foundation). Finally, the third tweet is more of a socio-political commentary espousing the misleading claim. These examples illustrate that it is important to extract the most representative set of keywords for any claim to understand the wide variety of ways it can be discussed.

Problem with existing Keyword Identification techniques. Several approaches to extract important keywords from social network posts have been proposed [10, 28, 46, 57]. We initially experimented with various existing keyword detection methods [10, 28] to extract the best set of keywords from the claims extracted in LAMBRETTA's previous step. Our analysis identified three main issues with previous approaches. We include three examples below obtained while testing with Yake [10]:

- Problem Type 1: Missing key entities
  Claim: Michigan Governor Whitmer send health dept into Detroit TFC Center to evict GOP poll-watchers but not Dem pollwatchers
  Automatically detected keywords using YAKE: *Detroit, TFC ,pollwatcher*
- Problem Type 2: High Recall, Low Precision
  Claim: Signature verification system in Clark County have 89% failure rates for catching poor signature matches
  Automatically detected keywords using YAKE: *signature, match*
- Problem Type 3: Low Recall, High Precision
  Claim: Tens of thousands of votes illegally received after 8 P.M. on Tuesday, Election Day
  Automatically detected keywords using YAKE: *ten,thousand,vote,8*

In the first example, the keyword extractor misses the key entities of the claim being discussed, *Whitmer* and *GOP*, and may produce many unrelated results. In the second example, the extracted keywords do capture the key entities, *signature* and *match*, but these keywords are generic and may result in many hits that are part of the larger narrative around signature matching on mail-in ballots rather than the specific claim of the failure rate of signature matches in Clark County. Finally, the third example extracts overly specific keywords that will miss relevant content.

The issue with existing approaches is that they identify the most important keywords by only looking at the claim body (e.g., the tweet already moderated by Twitter), without taking into account the context in which the claim is discussed. To overcome this issue, LAMBRETTA formulates the problem as a Learning to Rank (LTR) task [11], where the selection of optimal keywords from the claim is driven by the document store (i.e., $\mathbf{D}_1$, $\mathbf{D}_2$, etc.) containing examples of social media discussion of the claim in question. We note that we use both $\mathbf{D}_1$ and $\mathbf{D}_2$ in our experimental setup to cross-evaluate the LTR model trained on one document store and tested on an unseen

document store. We provide more details on the LTR task and our keyword detection experiment below.

Learning to Rank. Learning to Rank (LTR) is a task aiming to learn a function to rank the effectiveness of query terms in retrieving relevant documents from a document store. LTR based methods have been used in many information retrieval oriented applications [38, 44, 50, 87, 95]. We refer readers to [11] for a detailed review of the LTR task and different approaches to it.

To train our LTR model, we first annotate ground truth keywords for a fraction of the 900 misleading claims extracted from $\mathbf{G}_1$, manually labeling relevant tweets resulting from querying the keywords. We then develop features to train the LTR model and evaluate it on unseen tweets from $\mathbf{D}_1$ and $\mathbf{D}_2$. We describe these steps in detail below.

Building ground truth annotations. We begin by randomly selecting 125 misleading claims from our filtered set of claims (see Section III-A) to train our LTR model. We refer to these claims as ***archival train claims***. For each claim, we annotate the ground truth to be the set of keywords made up of terms from the misleading claims which produce the most related set of results when queried against our data store, optimizing for both relevance and size of results.

We start by initializing ***base keywords***, which are two words consisting of the subject and the object of a misleading claim. We query the two different Twitter data stores $\mathbf{D}_1$ and $\mathbf{D}_2$ with the ***base keywords*** and retrieve a set of results. As expected, the ***base keywords*** are usually pretty broad and return many irrelevant false positives. We fine tune the query by checking a random sample of 20 posts from query results and then adjust the ***base keywords*** by either adding new words from the query or removing words that are causing the false positives. We repeat the process until we find the most relevant set of keywords for each misleading claim. Finally, for each claim, the best set of keywords is tagged as a positive instance with the remaining keywords tagged as negative instances of relevance.

This ground truth is then used by LAMBRETTA to learn the ranking function that automatically identifies the best set of keywords for any given claim. This enables LAMBRETTA to automatically extract keywords from the body of a claim without any further external context of human intervention.

Data pre-processing. Before further analysis, LAMBRETTA removes stopwords from the tweet claims while doing basic pre-processing, e.g., lowercasing text and removing punctuation marks. We keep numbers since they can often be an integral part of extracted claims. We split the pre-processed misleading claims into n-grams of length two, three, four, and five, which are our potential set of query terms.

Feature Engineering. Next, we use the potential query terms returned by the previous step to extract our learning to rank model features. To this end, we query the two data stores $\mathbf{D}_1$ and $\mathbf{D}_2$, retrieving all the posts matching the query terms. LTR requires us to generate a dataset consisting of a query set, relevance information, and feature values to learn the ranking. Feature values can be a few or numerous. Applications of LTR have used features like document TF-IDF, BM25 [75] scores, document length, number of matching query terms, and number of query terms in important sections of a document, e.g., the title of a Web page [3, 53, 96]. The most widely used benchmark to build models based on LTR is the LETOR dataset [74], which contains query sets, learning features, and labeled rankings related to the 25 million page *GOV2* Web page collection [73]. Unfortunately, we cannot directly use the LETOR dataset since our data store is composed of posts from Twitter, which are fundamentally different from Web pages. Instead, we take inspiration from the LETOR dataset and develop six features that we use in our LTR experiments. In the following, we briefly discuss these features.

We use the term ***spanning subset*** to describe the earliest 20%, most recent 20%, and 10% of the middle-aged results returned from the query, based on the timestamp attached to their tweets. From this, we derive six features:

1) Total number of hits (matching tweets) produced.
2) Mean and median pairwise similarity score between the entries of the ***spanning subset***.
3) Mean and median similarity score between the entries in ***spanning subset*** and the claim.
4) Mean and median similarity score between the query and the claim.
5) Mean and median value of the TextRank scores [57] of the query terms.
6) Mean and median score of the Term Frequency-Inverse Document Frequency scores [75] of the query terms.

We need to extract ***spanning subset*** from the set of returned results as the number of results retrieved by the candidate keywords from the candidate query set might grow large, specially in cases of generic set of keywords in a query. Performing a pairwise semantic similarity comparison on this large set is not computationally feasible. The intuition behind ***spanning subset*** is to sample tweets during the different period of a discussion (from the onset to the current phase) along a timeline, compared to randomly sampling tweets. The similarity score between the set of results, and between the query and the results are calculated using the cosine similarity of the sentence embeddings encoded using a pre-trained *all-mpnet-base-v2* model proposed in [91]. The *all-mpnet-base-v2* model produces sentence embeddings with 768 dimensions, and had the best average performance on encoding sentences over 14 diverse tasks from different domains. Note that the features used in training the LTR model are not domain-dependent (i.e., election misinformation in this case), and are designed to entirely capture the semantic relatedness between query results and subsequent claims. These type of features will be useful for applying LAMBRETTA in other contexts for content moderation in different topics.

Training the LTR model. We use the RankLib project, part of the Lemur Toolkit [63] which includes a variety of LTR algorithms. While there are LTR algorithms that use complex neural architectures like Deep Learning [13, 69], we cannot

directly use them since they are designed to work on large scale datasets like LETOR. Instead, our LTR models are powered by features utilizing state of the art neural semantic models (*all-mpnet-base-v2*) to capture the interaction between query terms and result set.

We use all eight algorithms implemented by Lemur (MART, RankNet, RankBoost, AdaRank, Coordinate Ascent, LambdaMART, ListNet, and Random Forests) for our experiments. To evaluate our model, we use a rank-based evaluation metric commonly used in information retrieval settings: Mean Average Precision (MAP). In our case, our problem setting is a binary judgement where a keyword is either relevant to the misleading claim or not.

As with all keyword extraction problems, our dataset has many irrelevant combinations (majority class) of wrong keywords compared to one or two correct combinations (minority class) of optimal keywords we want to extract. This imbalance causes the learning algorithm to perform very poorly. To address this problem, prior work on LTR used BM25 as a pre-ranker to retrieve a small set of highly ranked documents from the entire document index before applying the LTR algorithm on the small subset of highly relevant documents [50]. We employ a similar pre-ranking step to filter out the set of keywords that retrieve irrelevant results for the claim by using a heuristic based on semantic query similarity. The idea behind the heuristic is that candidate keywords most likely to be optimal return tweets that are more semantically similar than unrelated ones. Thus, for each claim, its corresponding set of candidate keywords, and the retrieved results using these keywords as queries, we construct a subset called **FilteredQuerySet**, which is a ranked list of the top k results retrieved by all candidate keywords, sorted by the cosine similarity with the claim under question. The results returned by an irrelevant query have lower semantic similarity with the claims, thus failing to appear in the ranked set **FilteredQuerySet**. We experiment on different values of k and find that setting k to 20 includes all of the ground truth queries for training, validation, and test claims, while reducing the size of irrelevant candidate query set by 30 times.

To test our LTR model, we perform a 5-fold cross-validation on our ground truth. We observed that Random Forest with LambdaMART [8] as the bagging ranker produced the best results among the eight different ranking algorithms. We further increase performance by performing a grid search over hyper-parameters (number of leaves, number of bags, number of trees, and minimum leaf support). The 5-fold cross-validation on our ground truth using the tuned Random Forest model achieves a MAP of 0.768. As we will show later in Section IV-A, this convincingly outperforms other state-of-the-art keyword extraction approaches. We further refer to this LTR model as ***initial train model***.

Validating the LTR model. The previous experiment showed that our LTR approach can produce an accurate model on our training set. We now want to understand whether our model generalizes and can effectively identify posts related to claims that are not in the training set. To this end, we design and conduct two experiments. In the first experiment, we randomly select 75 additional misleading claims (extracted as per Section III-A) not part of the ground truth set, referring to them as ***archival validation claims***. The output keywords for these new claims can be inferred from the previously trained model, but they still require ground truth annotation to evaluate results. We thus manually label the ***archival validation claims*** via the same iterative method that we used for the ***archival train claims***.

Next, we generate the potential query terms set for the ***archival validation claims***, following the same steps for ***archival train claims***. We then query $\mathbf{D}_1$ and generate the feature values for candidate keywords for each claim. Finally, we perform inference on these new claims to see if our trained model can identify the best set of keywords. Our model achieves a MAP of 0.781, indicating that our approach effectively identifies keywords and retrieves data for previously unseen claims. After the validation step, we now have ground truth of 75 additional misleading claims from ***archival validation claims***, which we use to expand our overall training set to a total of 200 claims, which we call ***expanded claims***. We train a new LTR model on the ***expanded claims***, which we refer to as ***expanded model***. At this point, we have 700 claims remaining which were not manually annotated and are missing corresponding keywords. We later use the ***expanded model*** to extract the keywords for these claims.

In the second experiment, we aim to verify that the learned model is not biased towards the data store it was trained on (i.e., $\mathbf{D}_1$) and can be applied to an unseen data store ($\mathbf{D}_2$). In the previous experiments we generated the features from results queried on $\mathbf{D}_1$ for validation. Instead, in this experiment we build the features from results queried on $\mathbf{D}_2$. We re-use the ***expanded claims*** as our claim list, along with the corresponding ground truth we had collected for the previous training/validation experiments. Following the same steps as for the ***archival train claims*** and ***archival validation claims***, we generate the feature values for each of the candidate keywords for the claim by querying the datastore of $\mathbf{D}_2$. We use the previously trained ***expanded model*** model for inference on the ***expanded claims*** and achieve a MAP of 0.767, showing that our LTR model is not dependent on the datastore it was trained on.

C. Data store retrieval

After training our LTR model and validating its performance on ranking experiments with other methods, we apply the ranking model to the remaining 700 misleading claims. The output is 499 unique sets of keywords. This number is lower than the total number of claims (900) since the keywords extracted from two different claims can be the same. We use these 499 sets of keywords to query the two data stores $\mathbf{D}_1$ and $\mathbf{D}_2$. We require candidates to match only if all keywords in a query are present in a tweet, regardless of the order of the tokens. We also make sure to exclude retweets and quoted tweets when searching for the relevant tweets.

After searching for the keywords, we obtain 2,042,173 tweets from $\mathbf{D}_2$ and 101,353 tweets from $\mathbf{D}_1$. The average number of tweets per misleading claim from $\mathbf{D}_2$ is 5,988 and for $\mathbf{D}_1$ it is 203.11. We use the 101,353 tweets from $\mathbf{D}_1$ to further evaluate LAMBRETTA throughout the rest of the paper, as checking the tweets flagged for moderation from $\mathbf{D}_2$ is not feasible due to Twitter API limitations.

IV. EVALUATION

To evaluate LAMBRETTA, we first compare the quality of the candidate tweets extracted by our system to those recommended by other state-of-the-art approaches. Next, we manually analyze a subset of the tweets flagged by LAMBRETTA to assess its false positives and false negatives. We then check whether the tweets flagged by LAMBRETTA received soft moderation from Twitter, in the context of the 2020 US Election voter fraud allegations, finding that only a small fraction of them did. Finally, given the disparity between the soft moderation candidates flagged by LAMBRETTA and those that received labels by Twitter, we use our dataset to understand if Twitter's moderation is driven by specific characteristics of the tweets or of the users posting them.

A. Comparison with other keyword extraction and information retrieval based methods

In Section III-B, we showed some examples of why existing keyword extraction methods might be unsuitable for our task, motivating us to develop the LTR component of LAMBRETTA. We now provide a rigorous quantitative analysis of this fact, by comparing the LTR model used by LAMBRETTA with three other keyword extraction algorithms: YAKE [10], KeyBERT [28], and RAKE [79]. As an alternative to keyword extraction algorithms, another possible approach to finding similar tweets given a source one is leveraging semantic search techniques [30]. We also evaluate state-of-the-art methods in this space against LAMBRETTA's LTR model; more precisely semantic search using Sentence Transformers [77] and BM25 [78], which can be used to get tweets matching a query tweet by using a ranking function.

To establish ground truth, we sample 60 random claims from the set of 200 ***expanded claims***. For these sets of claims and the ground-truth keywords, we manually verify that each tweet returned by the keywords does discuss the claims in question and filter out any irrelevant ones. This yields a set of 10,776 tweets associated with the 60 claims. Sentence Transformers and BM25 also require tuning a similarity threshold for matching and ranking tweets; for these experiments, we retrieve tweets using different thresholds, ranging from 0.3 to 0.9, and select the threshold for each method that achieves the highest F1 score based on our ground truth.

Figure 3 reports the F1 score for all methods, including LTR, when extracting similar tweets to the 60 claims curated from ***expanded claims***. LAMBRETTA's LTR is the best performing model, with over 60% of the claims having an F1 score of 0.8 or higher. The second best performing algorithm is YAKE, with less than 40% of claims having an F1 score of at least 0.8. BM25 and semantic search using Sentence embeddings perform poorly, as does KeyBERT. These results confirm that the LTR model, the keyword identification component of LAMBRETTA, is effective at identifying similar tweets with better Precision and Recall than other keyword extraction and semantic search based methods.

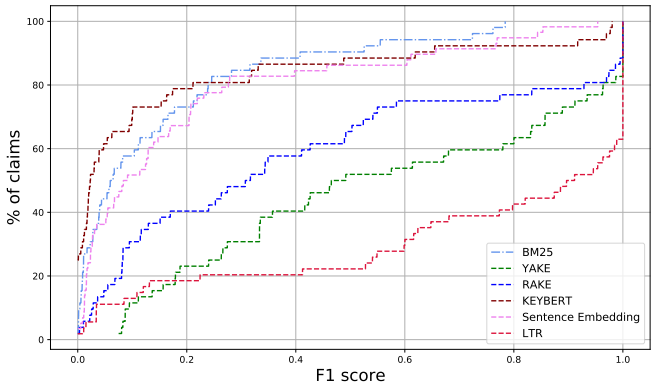


Figure 3: Performance of the LTR model vs. other keyword identification and semantic search methods.

Comparison of the reduction of effort offered by different methods to content moderators. The goal of LAMBRETTA is to provide a set of social network messages to content moderators, allowing them to make informed decisions and to keep the platform safe. To be useful and avoid overwhelming content moderators, this system should reduce the number of candidates as much as possible, while at the same time maximizing Precision and Recall. From our comparison experiments described above, we find that LAMBRETTA reduces the pool of tweets which should be checked for moderation compared to all other systems. LAMBRETTA retrieves 5.33 times less tweets related to a moderated claim than the second best performing system, YAKE. This shows that LAMBRETTA is better suited than alternative approaches for the task of aiding moderation of misleading information on social media.

B. Validation of LAMBRETTA

In this section, we first perform a manual analysis on the results recommended by LAMBRETTA, assessing its False Positives and False Negatives. We then check if the recommendations made by LAMBRETTA were also soft moderated by Twitter, finding that only a small fraction of tweets flagged by LAMBRETTA were intervened by the platform.

Manual validation. To understand the quality of the recommendations made by our approach, we perform a manual examination of the tweets recommended by LAMBRETTA, aiming to check if they should indeed have been moderated. The first author of this paper samples 1,500 tweets among the candidates flagged by LAMBRETTA and analyzes them qualitatively, identifying seven categories that they can fall under: 1) Amplyfing tweets, 2) Reporting Tweets, 3) Counter Tweets, 4) Satire Tweets, 5) Discussion Tweets, 6) Inquiry Tweets, 7) Irrelevant Tweets. The process goes through multiple iterations of coding the sample tweets, grouping them

into different categories. The annotator then consolidates overlapping themes and categories, converging to seven which we discuss below. Note that a single tweet can fall under multiple categories, e.g., they can amplify a misleading claim and prompt a discussion at the same time. The definition of each category, alongside a representative example tweet of the category is presented below:

1) Amplifying Tweet: it positively reinforces the misleading claim and aims to further spreading the message.

> "Terry Mathis (born 1900) apparently voted via absentee ballot in Wayne County: Michigan. It doesn't stop here. This person applied for an absentee ballot on December 2: the ballot was then sent out AND returned in the same day.<URL> <URL>"

2) News Reporting: it reports the headline of a misleading news article or another tweet, without any additional commentary and text from the tweet's author.

> "BREAKING: Unofficial: Trump trailing Biden by only 4,202! There is a ballot count upload glitch in Arizona. Reports saying over 6,000 False Biden Votes Discovered <URL>."

3) Counter Claim: it attempts to question and/or debunk the misleading information.

> "Misleading claims that Trump ballots in Arizona were thrown out because Sharpie pens were provided to voters are untrue. A ballot that cannot be read by the machine would be re-examined by hand and not invalidated if it was marked with a Sharpie.#Election2020 <URL>."

4) Satire: it discusses the false claim in a satirical way.

> "@<USER> He was allegedly slain by Soros, who then had Chavez's personal army of false voters cram him inside a Dominion voting machine before loading him into an RV with Hunter Biden's second laptop and Hillary's server."

5) Discussion: it prompts discussion of the details of the misleading claim by adding commentary.

> "@<USER> What happened to all the votes cast for Trump that were destroyed? How are those tallied? Detroit-based Democratic Party activist a local: boasts On FB: I threw out every Trump ballot I saw while working for Wayne County, Michigan. They number in the tens of thousands, as did all of my coworkers."

6) Inquiry: it inquires about the details of events related to the misleading claim and does not attempt to either support or deny the claim under question.

> "Has anyone got a compelling justification for this? In accordance with a tweet I saw from @<USER>: A "James Bradley" born in 1900 has recently been entered into the Michigan Voter Information Center. James apparently submitted an absentee ballot on October 25. For a 120-year-old, not bad! <URL>"

Category	Candidates	Moderated (%)
Amplifying	1,198	241 (20.11%)
Reporting	922	222 (24.07%)
Counter	122	4 (3.27%)
Satire	15	0 (0.00%)
Discussion	646	83 (12.84%)
Inquiry	84	22 (26.19%)
Irrelevant	59	1 (1.69%)

Table I: Categories of candidate tweets and number/percentage receiving soft moderation by Twitter.

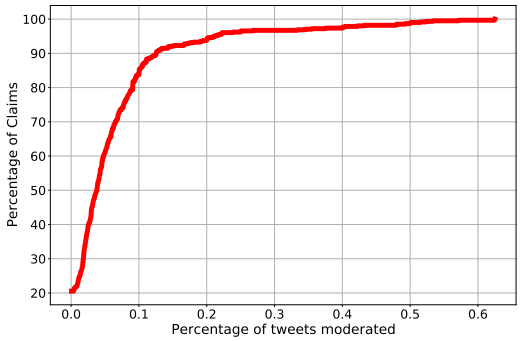


Figure 4: Moderation coverage of misleading tweets flagged by LAMBRETTA per claim.

7) Irrelevant: it is irrelevant to the misleading claim. These are considered false positives.

> "#LOSANGELES: Our truck will be at the @Hollywood-Bowl voting location till 7pm. Use your voting rights and reward yourself with some."

Table I, reports the number of candidate tweets in each category. The vast majority falls in the amplification category with 1,198 out of 1,500 tweets (79.86%), followed by tweets reporting about the false claim with 922 tweets (61.46%). 43% of the tweets add further discussion to the misleading claim under question rather than simply sharing the headline of a news article, and 8% of them try to debunk it. Finally, 59 (3.93%) of the tweets flagged by LAMBRETTA are irrelevant to the claim under study and can therefore be considered false positives. As mentioned, the goal of LAMBRETTA is to flag tweets that are related to a claim that the platform wants to moderate, but human moderators should still make the final decision about applying labels to the candidates flagged by our system. We further discuss the implications of running LAMBRETTA in the wild in Section VI.

False Negatives. To evaluate the False Negatives of LAMBRETTA, we first evaluate the false negative of each of its two phases separately using $\mathbf{G}_1$ for the first phase, and $\mathbf{D}_1$ for the second phase. In the claim structure extraction module, the Proposition Extractor component fails to extract 2.77% of the propositions that are claim span. After the propositions are extracted, LAMBRETTA misclassifies 3.46% of the propositions that contain a claim, implying the missed claim structure would not be processed further in the second phase.

In the second phase, we quantify the proportion of tweets missed by the keywords identified through the LTR component of LAMBRETTA. The keywords produced by LTR identify 8,748 of the 10,776 tweets in the ground truth; this yields an 18.81% false negative rate from LAMBRETTA's keyword extraction phase. This is much lower than the false negative rate of the second best state-of-the-art approach, YAKE, which is 32.45%.

Comparison to Twitter's soft moderation. After determining that the recommendations made by LAMBRETTA are accurate, we check if the tweets recommended by our approach were also soft-moderated by Twitter. For every claim from the Claim Extraction Module, we retrieve the relevant set of tweets guided by the best set of keywords from our LTR component. We then follow [104] and extract metadata of soft moderation interventions for each tweet (i.e., if the tweet received a soft moderation and the corresponding warning label). We perform this experiment on $\mathbf{D}_1$.

Out of the 101,353 tweets flagged by LAMBRETTA as candidates for moderation, we find that only 4,330 (4.31%) were soft moderated by Twitter. Note that we could not check the existence of warning labels for 993 tweets as they were inaccessible, with either the tweets having been deleted or the accounts that posted them being deleted or suspended. This experiment highlights the limitations of Twitter's soft moderation approach, suggesting that the platform would benefit from an automated system like LAMBRETTA to aid content moderation. In Section IV-C, we further investigate whether we can identify a specific strategy followed by Twitter in moderating content.

C. What drives Twitter moderation?

The analysis from the previous sections shows that Twitter only moderates a small fraction of tweets that should be moderated. In this section, we aim to better understand how these moderation decisions are made.

We start by examining whether certain claims are moderated more aggressively than others and whether the type of message in a tweet affects its chances of being moderated. We then analyze the text and the URLs in moderated and unmoderated tweets, aiming to ascertain: 1) whether Twitter uses text similarity to identify moderation candidates and 2) whether Twitter automatically moderates all tweets linking to a known misleading news article. Next, we look at the account characteristics of the users who posted moderated and unmoderated tweets, and engagement metrics (i.e., likes and retweets), aiming to understand if Twitter prioritizes moderating tweets by popular accounts or viral content.

Coverage by claim. In Figure 4, we plot the Cumulative Distribution Function (CDF) of the percentage of tweets moderated by Twitter for each of our 900 claims, out of the total candidate set flagged by LAMBRETTA. Approximately 80% of the claims have less than 10% of the tweets moderated, whereas 95% of claims have close to 20% of the tweets moderated. Very few claims (5) have at least half of the tweets moderated. The misleading claim with the highest coverage is "***Russ Ramsland file affidavit showing physical impossibility of election result in Michigan***" with 159 out of 309 (51%) candidate tweets receiving moderation labels by Twitter. On the other hand, the claim "***Chinese Communists Used Computer Fraud and Mail Ballot Fraud to Interfere with Our National Election***" only has 1 out of 236 tweets (0.42%) with warning labels. This shows that, while the fraction of pertinent tweets moderated by Twitter is generally low, the platform seems to moderate certain claims more aggressively than others.

Coverage by tweet type. In Section IV-B, we list seven categories of tweets discussing misleading claims. We now set out to understand whether Twitter moderates certain types of tweets more than others. Table I shows the fraction of tweets in our sample set of 1,500 manually analyzed tweets that did receive soft moderation by Twitter, broken down by category. Tweets raising questions, reporting, or amplifying false claims are more likely to be moderated (with 26.19%, 24.07%, and 20.11% of their tweets being moderated, respectively). Satire tweets never received moderation labels, while tweets debunking false claims were only moderated in 3.27% of the cases. This indicates that Twitter considers the stance of a tweet mentioning a false claim, perhaps as part of a manual moderation effort.

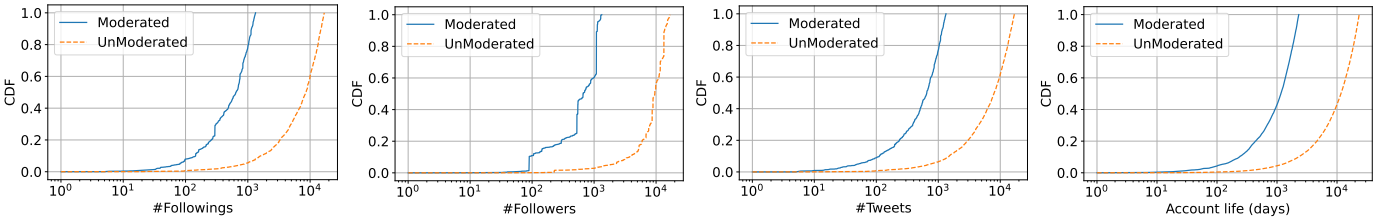
Content analysis. Next, we investigate whether Twitter looks at near identical tweets when applying soft moderation decisions. We take all tweets flagged as candidates by LAMBRETTA, and group together those with a high Jaccard similarity of their words. We remove all the links, user mentions, and lemmatize the tweet tokens by using the ***ekphrasis*** tokenizer [4]. We consider two tweets to be near identical if their Jaccard similarity is in the range 0.75–0.9 (out of 1.0). We do so to extract tweet pairs that are not exactly the same, but have some variation in the content while discussing the same misleading claim. We exclude retweets, and only consider the tweets originally authored by the users.

We extract 17,241 pairs of tweets (out of 438,986 possible pairs), where at least one of the two was moderated by Twitter. Only 3,857 pairs have both tweets moderated. Note that LAMBRETTA effectively identifies *all* the 17,241 pairs of tweets as moderation candidates. Here is an example of a very similar pair of tweets, for which Twitter did not add labels to one of them:

> **Moderated**: "RudyGiuliani in Trump campaign news conference: ""Joe Biden said a few weeks ago that his voting fraud crew was the best in the world. They were excellent, but we got them!""
> **Unmoderated**: "Joe Biden said a few weeks ago that his crew was the greatest in the world at catching voter fraud, but we caught them."

These findings indicate that the decision by Twitter to add soft moderation to a tweet does not seem to be driven by the lexical similarity of tweets.

URL analysis. Another potential indicator used by Twitter when deciding which tweets to moderate is whether they

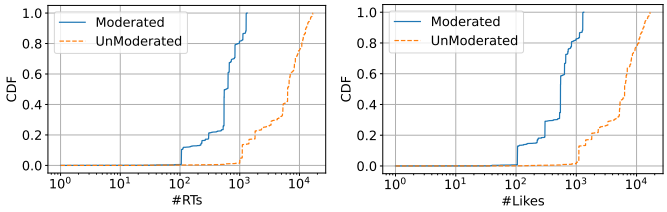


(a) Following **(b)** Followers **(c)** Tweet Counts **(d)** Account Age

Figure 5: Cumulative Distribution Functions (CDF) of various user metrics for moderated and unmoderated tweets.

URL news story	Candidates	Moderated
USPS whistleblower	315	7 (2.2%)
China manipulating election	252	4 (1.5%)
Michigan ballot dump	215	44 (20%)
#Suitcasegate related FB video	208	3 (1.4%)
Dominion remote machine control	135	15 (11%)

Table II: Examples of URLs in candidate tweets and those being moderated by Twitter.



(a) Retweets **(b)** Likes

Figure 6: CDFs of engagement metrics for moderated and unmoderated tweets.

include links to known news disinformation articles. First, we expand all the links in the body of candidate tweets identified by LAMBRETTA to get rid of URL shorteners [54]. This yields 13,108 distinct URLs. Next, we group candidate tweets by URLs and check what fraction of tweets sharing the URL are moderated by Twitter.

Table II shows the five most common URLs (abstracted to the topic of the news articles) in our dataset, with the fraction of tweets including those URLs moderated by Twitter. All these news stories, excluding one Facebook video, originate from known low-credibility websites like TheGatewayPundit and DC Dirty Laundry, which promote election misinformation. Twitter moderates tweets containing those URLs in an inconsistent matter. Also note that LAMBRETTA can help identify 4,598 additional moderation candidate tweets compared to those on which Twitter intervened.

User analysis. We examine the differences in the social capital (e.g., number of followers) of the authors of tweets moderated by Twitter, compared to those our system recommends for moderation but for which Twitter did not intervene. Figure 5 reports the CDF of followers, following, tweet count, and account age of accounts that posted moderated and unmoderated tweets. We find that authors of tweets that have warning labels have much fewer followers, followings, lower account activity, and have younger accounts than tweets without warning labels. We also conduct two-sample Kolmogorov-Smirnov tests for each user metric, finding that the differences are statistically significant for followers and account age ($p < 0.01$) as well as following count and status count ($p < 0.05$). This goes against the notion that popular accounts are more likely to have their content moderated.

We also check if the accounts with moderated tweets were suspended for violating Twitter Rules [100]. We find that only 33 out of 3,397 users were suspended by Twitter; this gives us strong ground to rule out the possibility that tweet moderation is not due to the "legitimacy" of the account themselves.

Engagement analysis. Finally, we analyze engagement metrics. Figure 6 reports the CDF of retweets and likes categorized by moderation status of the 101,353 candidate tweets LAMBRETTA flags for moderation from $\mathbf{D}_1$, compared to the ones flagged by Twitter. Similar to the user analysis, we find that unmoderated tweets have more engagement. When we check for statistical significance of difference in distributions of the retweet count using Kolmogorov-Smirnov tests, we find that it is statistically significant ($p < 0.01$), while we cannot reject the null hypothesis for the likes. Note, however, that these results have to be taken with a grain of salt, as we do not have a timeline of when exactly moderation was applied, and whether the soft interventions hampered the virality of online content.

Takeaways. Our analysis paints a puzzling picture of soft moderation on Twitter. We find that certain claims are moderated more aggressively. Still, Twitter does not seem to have a system in place to identify similar tweets discussing the same false narrative, nor flagging tweets that link to the same debunked news article. We also find that Twitter does not appear to focus on the tweets posted by popular accounts for moderation, but rather that tweets posted by accounts with more followers, friends, activity, and a longer lifespan are more likely to go unmoderated. This confirms the need for a system like LAMBRETTA.

V. RELATED WORK

In this section, we review relevant work on soft moderation, security warnings, and keyword extraction in the context of disinformation.

Soft Moderation during the 2020 Elections. As part of the Civic Integrity Policy efforts surrounding the 2020 US elections, Twitter applied warning labels on "misleading information." Empirical analysis [104] reports 12 different types of warning messages occurring on a sample of 2,244 tweets with warning labels. Statistical assessment of the impact of Twitter labels on Donald Trump's false claims during the 2020 US Presidential election finds that warning labels did not result in any statistically significant increase or decrease in the spread of misinformation [67]. Twitter later reported that approximately 74% of the tweet viewership happened post-moderation and, more importantly, that the warnings yielded an estimated 29% decrease in users quoting the labeled tweets [99].

Security Warnings for Disinformation. The warning labels adopted by Twitter as soft moderation intervention can be broadly categorized as a type of security warning. Security warnings can be classified into two types: contextual and interstitial. The former passively inform the users about misinformation through UI elements that appear alongside social media posts. The latter prompt the user to engage before taking action with the potential piece of disinformation (e.g., retweeting or sharing). A recent study [37] shows that interstitial warnings may be more effective, with a lower click-through rate of misleading articles. Additionally, interstitial warnings are more effective design-wise because they capture attention and provide a window of opportunity for users to think about their actions. Efforts to study warning labels on countering disinformation have thus far been mostly focused on Facebook [70, 71, 80], where warning labels were limited to "disputed" or "rated false", and the approach was deemed to be of limited utility by Facebook [90]. Recently, other platforms like Twitter [2], Google [41], and Bing [56] also used some form of fact-check warnings to counter disinformation.

Tools for automated content moderation. The sheer scale of content being produced on modern social media platforms (Facebook, Reddit, YouTube etc.) have motivated the need to adopt tools for automated content moderation [6, 26]. However, due to the nuanced and context-sensitive nature of content moderation, it is a complex socio-technical problem [34, 83]. Most of the work in this space of automated content moderation are focused on Reddit, aiming to identify submissions that violate community-specific policies and norms ranging from hate speech to other types of problematic content [82]. The most popular solution to automated content moderation in Reddit, AutoModerator [34] allows community moderators to set up simple rules based on regular expressions and metadata of users for automated moderation. On YouTube, FilterBuddy [35] is available as a tool for creator-led content moderation by designing filters for keywords and key phrases. Similarly, Twitch offers an automated moderation tool called Automod to allow creators to moderate four categories of content (discriminations and slurs, sexual content, hostility, and profanity) on the platform [59]. Another tool, called CrossMod [12] uses an ensemble of models learned via cross-community learning from empirical moderation decisions made on two subreddits of over 10M subscribers each.

Keyword Extraction for Disinformation. Researchers have used an array of methods to detect disinformation, ranging from modeling user interactions [72, 88, 98], leveraging semantic content [16, 18, 66, 107], and graph based representations [23, 51, 62]. The foundation of our system lies in the keyword detection, which has been used before to study disinformation on social media. DisInfoNet, a toolbox presented in [29], represents news stories through keyword-based queries to track news stories and reconstruct the prevalence of disinformation over time and space. Similarly, the work in [22] uses keyword extraction techniques as the base for semantic search to detect fake news on WhatsApp. The work in [14] focuses on credibility assessment of textual claims on news articles with potentially false information, also using keyword extraction as a part of their multi-component module.

Learning To Rank for Keyword extraction. The closest applications of LTR to our work are the proposals in [36] for keyphrase extraction and in [9] for keyword extraction in Chinese news articles. The foundational work by [36] motivates the necessity of framing the problem of keyphrase extraction as a ranking task rather than a classification task while improving results on extracting keyphrases from academic research papers, and social tagging data. The LTR approach for keyword extraction utilized by LAMBRETTA is motivated by the premise set up by this work. Similarly, [9] use Learning To Rank to identify keywords from 1800 public Chinese new articles using TF-IDF, TextRank, and Latent Dirichlet Allocation (LDA) as the set of features for the ranking model.

VI. Discussion and Conclusion

This paper presented LAMBRETTA, a system geared to automatically flag candidate tweets for soft moderation interventions. Our experiments demonstrate that Learning to Rank (LTR) techniques are effective, that LAMBRETTA outperforms other approaches, produces accurate recommendations, and can increase the set of tweets that receive soft moderation by over 20 times compared to those flagged by Twitter during the 2020 US Presidential Election.

Implications for social media platforms. As discussed in Section IV-C, soft moderation interventions applied by Twitter appear to be spotty and not following precise criteria. This might be due to moderation being conducted mainly in an ad-hoc fashion, relying on user reports and the judgment of moderators. LAMBRETTA can assist this human effort, working upstream of the content moderation process and presenting moderators with an optimal set of tweets that are candidates for moderation. Because of the nuances of moderating false and misleading content, we envision LAMBRETTA to be deployed as an aid to human moderation rather than an automated detection tool.

Nonetheless, the claim-specific design of LAMBRETTA can also be used by moderators for other actions as per their

policies, e.g., asking users to remove a given tweet or performing hard moderation by removing the tweets. The choice between soft moderation and hard moderation can be made by moderators contextually after the moderation candidates are retrieved through LAMBRETTA, either based on underlying claims or a case-by-case basis. E.g., platforms may decide to soft moderate posts that push a certain false narrative but not add warnings if posts inform users about falsehood. Alternatively, they might add warnings to posts about the false narrative, providing additional context to users and allowing them to make up their minds about it. Platforms could also craft warning messages depending on the context in which a false claim is discussed or design these messages to be more effective based on the audience and risk levels of specific false claims. For example, different type of warning messages can be applied by platforms to distinguish between different levels of risk associated with the misleading claims (e.g., high and low-level risks associated with COVID-19 misinformation) [49]. We are confident that Human-Computer Interaction researchers will be able to address these challenges, which go beyond the scope of this paper.

Human effort required for adopting LAMBRETTA. When setting up LAMBRETTA to work in a new context, platform moderators need to follow the steps highlighted in Sections III-A and III-B. First, they need a set of tweets with claims they identified as containing misleading information, together with a tuning dataset like $\mathbf{D}_1$. Moderators can create a tuning dataset like $\mathbf{D}_1$ by using a broad set of keywords associated with the event or topic and querying the Twitter API to which they have full access (e.g. "COVID-19," "coronavirus," etc., in the case of the pandemic). They then need to tune the threshold for the Claim Stopper API in the Claim Extraction component (see Section III-A). In our experiments, this phase took us, on average, two minutes per claim. Finally, they need to create the training set for the LTR model by following the iterative process discussed in Section III-B. When performed by a single annotator, this process took, on average, 15 minutes per claim for the experiments discussed in this paper. Twitter could speed up these steps further by having multiple annotators work on the same task. Additionally, the work required on each claim is independent of other claims; therefore this process can be easily parallelized within the organization or even through crowdsourcing campaigns [21, 45, 64].

Resilience to evasion. As with any adversarial problem, malicious actors are likely to try to evade being flagged by LAMBRETTA. E.g., they might avoid using certain words to avoid detection and use synonyms or dog whistles instead [25, 97, 106, 109]. However, this would make the false messaging less accessible to the general public, who would need to first understand the alternative words used and ultimately be counterproductive for malicious actors by limiting the reach of false narratives.

Limitations. LAMBRETTA requires a seed of tweets to be moderated, making it inherently reactive. However, this is a problem common to all moderation approaches, including the work conducted by fact-checking organizations. Another limitation is that we could only test LAMBRETTA on one dataset related to the same major event (the 2020 US Presidential Election), as this is the only reliable dataset with soft moderation labels available to the research community.

Even though Twitter applied warning labels on misinformation about COVID-19, previous research reported that these were unreliable and inconsistent [43, 52], which we independently confirmed in our preliminary analysis. More recently, Twitter recently started applying warning labels to tweets in the context of the Russian invasion of Ukraine [5], but these labels are applied based on the account posting them (i.e., if the account belongs to Russian or Belarusian state-affiliated media) instead of being claim-specific as required by LAMBRETTA. While the LTR model used by LAMBRETTA is not specific to the actual keywords being searched, and therefore we expect that it should generalize across the entirety of Twitter, platform moderators using the tool should take further steps to validate it when used in contexts other than politics and elections.

Future work. We plan to extend LAMBRETTA to additional platforms. Since our system only needs the text of posts as input, we expect it to generalize to other platforms, e.g., Facebook, Reddit, etc. We will also investigate how claims automatically built by LAMBRETTA can be incorporated into warning messages to provide more context to users and allow them to be better protected against disinformation.

Acknowledgments. We thank the anonymous reviewers for their comments that helped us improve the paper. Our work was supported by the NSF under grants CNS-1942610, IIS-2046590, CNS-2114407, IIP-1827700, and CNS-2114411, and by the UK's National Research Centre on Privacy, Harm Reduction, and Adversarial Influence Online (REPHRAIN, UKRI grant: EP/V011189/1).

REFERENCES

[1] A. Abilov, Y. Hua, H. Matatov, O. Amir, and M. Naaman. VoterFraud2020: a Multi-modal Dataset of Election Fraud Claims on Twitter. In *AAAI International Conference on Web and Social Media (ICWSM)*, volume 15, 2021.

[2] D. Alba and K. Conger. Twitter moves to target fake videos and photos. https://www.nytimes.com/2020/02/04/technology/twitter-fake-videos-photos-disinformation.html, 2020.

[3] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, and K. Weinberger. Learning to rank with (a lot of) word features. *Information retrieval*, 13(3), 2010.

[4] C. Baziotis, N. Pelekis, and C. Doulkeridis. Datastories at semeval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada, 2017. Association for Computational Linguistics.

[5] S. Benson. Twitter to label all State-Affiliated Russia media. https://www.politico.com/news/2022/02/28/twitter-label-state-affiliated-russia-media-00012351, 2022.

[6] M. Bickert. Publishing our internal enforcement guidelines and expanding our appeals process. https://about.fb.com/news/2018/04/comprehensive-community-standards/, 2018.

[7] S. Bradshaw and S. Grossman. Were Facebook and Twitter consistent in labeling misleading posts during the 2020 election? https://www.lawfareblog.com/were-facebook-and-twitter-consistent-labeling-misleading-posts-during-2020-election, 2022.

[8] C. J. Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581), 2010.

[9] X. Cai and S. Cao. A keyword extraction method based on learning to rank. In *2017 13th International Conference on Semantics, Knowledge and Grids (SKG)*. IEEE, 2017.

[10] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt. Yake! collection-independent automatic keyword extractor. In *European Conference on Information Retrieval*. Springer, 2018.

[11] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *International Conference on Machine Learning (ICML)*, 2007.

[12] E. Chandrasekharan, C. Gandhi, M. W. Mustelier, and E. Gilbert. Crossmod: A cross-community learning-based system to assist reddit moderators. *CHI Conference on Human Factors in Computing Systems*, 3(CSCW), 2019.

[13] M. Chen and X. Zhou. DeepRank: Learning to rank with neural networks for recommendation. *Knowledge-Based Systems*, 209, 2020.

[14] N. Choudhary, R. Singh, I. Bindlish, and M. Shrivastava. Neural network architecture for credibility assessment of textual claims. *arXiv:1803.10547*, 2018.

[15] N. Christin, S. S. Yanagihara, and K. Kamataki. Dissecting one click frauds. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2010.

[16] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini. Computational fact checking from knowledge networks. *PloS one*, 10(6), 2015.

[17] L. Del Corro and R. Gemulla. Clausie: clause-based open information extraction. In *The Web Conference (WWW)*, 2013.

[18] S. Esmaeilzadeh, G. X. Peh, and A. Xu. Neural abstractive text summarization and fake news detection. *arXiv:1904.00788*, 2019.

[19] O. Etzioni, A. Fader, J. Christensen, S. Soderland, et al. Open information extraction: The second generation. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[20] C. Fiesler and N. Proferes. "Participant" perceptions of Twitter research ethics. *Social Media+ Society*, 4(1), 2018.

[21] A. M. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and N. Kourtellis. Large scale crowdsourcing and characterization of twitter abusive behavior. In *AAAI International Conference on Web and Social Media (ICWSM)*, 2018.

[22] J. Gaglani, Y. Gandhi, S. Gogate, and A. Halbe. Unsupervised whatsapp fake news detection using semantic search. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2020.

[23] S. C. R. Gangireddy, C. Long, and T. Chakraborty. Unsupervised fake news detection: A graph-based approach. In *Proceedings of the 31st ACM conference on hypertext and social media*, 2020.

[24] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *arXiv:1803.07640*, 2018.

[25] Y. Gerrard. Beyond the hashtag: Circumventing content moderation on social media. *New Media & Society*, 20(12), 2018.

[26] Google. Youtube community guidelines enforcement in Google's transparency report for 2018. https://transparencyreport.google.com/youtube-policy/removals, 2018.

[27] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @ spam: the underground on 140 characters or less. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2010.

[28] M. Grootendorst. KeyBERT: Minimal keyword extraction with BERT. https://doi.org/10.5281/zenodo.4461265, 2020.

[29] S. Guarino, N. Trino, A. Chessa, and G. Riotta. Beyond fact-checking: Network analysis tools for monitoring disinformation in social media. In *International conference on complex networks and their applications*. Springer, 2019.

[30] R. Guha, R. McCool, and E. Miller. Semantic search. In *The Web Conference (WWW)*, 2003.

[31] N. Hassan, F. Arslan, C. Li, and M. Tremayne. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2017.

[32] G. Ho, A. Sharma, M. Javed, V. Paxson, and D. Wagner. Detecting credential spearphishing in enterprise settings. In *USENIX Security Symposium*, 2017.

[33] C. Iglesias Keller. Don't Shoot the Message: Regulating Disinformation Beyond Content. *Direito Público*, 2021.

[34] S. Jhaver, I. Birman, E. Gilbert, and A. Bruckman. Human-machine collaboration for content regulation: The case of reddit automoderator. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 26(5), 2019.

[35] S. Jhaver, Q. Z. Chen, D. Knauss, and A. X. Zhang. Designing Word Filter Tools for Creator-led Comment Moderation. In *CHI Conference on Human Factors in Computing Systems*, 2022.

[36] X. Jiang, Y. Hu, and H. Li. A ranking approach to keyphrase extraction. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009.

[37] B. Kaiser, J. Wei, E. Lucherini, K. Lee, J. N. Matias, and J. Mayer. Adapting security warnings to counter online disinformation. In *USENIX Security Symposium*, 2021.

[38] A. Karatzoglou, L. Baltrunas, and Y. Shi. Learning to rank for recommender systems. In *Proceedings of the 7th ACM Conference on Recommender Systems*, 2013.

[39] K. Klonick. The new governors: The people, rules, and processes governing online speech. *Harvard Law Review*, 2017.

[40] L. Konstantinovskiy, O. Price, M. Babakar, and A. Zubiaga. Toward automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. *Digital threats: research and practice*, 2(2), 2021.

[41] J. Kosslyn and C. Yu. Fact check now available in Google search and news around the world. https://blog.google/products/search/fact-check-now-available-google-search-and-news-around-world/, 2017.

[42] P. Krafft and J. Donovan. Disinformation by design: The use of evidence collages and platform filtering in a media manipulation campaign. *Political Communication*, 2020.

[43] J. Lange. Twitter is now flagging the use of 'oxygen' and 'frequency' in the same tweet, prompting new meme. https://theweek.com/speedreads/922275/twitter-now-flagging-use-oxygen-frequency-same-tweet-prompting-new-meme, 2020.

[44] R. Leaman, R. Islamaj Doğan, and Z. Lu. DNorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22), 2013.

[45] M. Lease and E. Yilmaz. Crowdsourcing for information retrieval. In *ACM SIGIR Forum*, 2012.

[46] S.-J. Lee and H.-J. Kim. Keyword extraction from news corpus using modified TF-IDF. *The Journal of Society for e-Business Studies*, 14(4), 2009.

[47] K. Levchenko, A. Pitsillidis, N. Chachra, B. Enright, M. Félegyházi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, H. Liu,

et al. Click trajectories: End-to-end analysis of the spam value chain. In *IEEE Symposium on Security and Privacy*, 2011.
[48] W.-Y. Lim, M.-L. Lee, and W. Hsu. ClaimFinder: A Framework for Identifying Claims in Microblogs. In *# Microposts*, 2016.
[49] C. Ling, K. P. Gummadi, and S. Zannettou. "Learn the Facts About COVID-19": Analyzing the Use of Warning Labels on TikTok Videos. *arXiv:2201.07726*, 2022.
[50] T.-Y. Liu et al. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3), 2009.
[51] Y.-J. Lu and C.-T. Li. GCAN: Graph-aware co-attention networks for explainable fake news detection on social media. *arXiv:2004.11648*, 2020.
[52] K. Lyons. Twitter promises to fine-tune its 5G coronavirus labeling after unrelated tweets were flagged. https://www.theverge.com/2020/6/27/21305503/twitter-labels-5g-conspiracy-coronavirus, 2020.
[53] C. Macdonald, R. L. Santos, and I. Ounis. On the usefulness of query features for learning to rank. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012.
[54] F. Maggi, A. Frossi, S. Zanero, G. Stringhini, B. Stone-Gross, C. Kruegel, and G. Vigna. Two years of short urls internet measurement: security threats and countermeasures. In *The Web Conference (WWW)*, 2013.
[55] K. Meng, D. Jimenez, F. Arslan, J. D. Devasier, D. Obembe, and C. Li. Gradient-based adversarial training on transformer networks for detecting check-worthy factual claims. *arXiv:2002.07725*, 2020.
[56] Microsoft Bing Blogs. Bing adds Fact Check label in SERP to support the ClaimReview markup. https://blogs.bing.com/Webmaster-Blog/September-2017/Bing-adds-Fact-Check-label-in-SERP-to-support-the-ClaimReview-markup, 2017.
[57] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.
[58] S. Mirza, L. Begum, L. Niu, S. Pardo, A. Abouzied, P. Papotti, and C. Pöpper. Tactics, threats & targets: Modeling disinformation and its mitigation. In *ISOC Network and Distributed Systems Security Symposium (NDSS)*, 2023.
[59] C. Moderation. How to Use AutoMod. https://help.twitch.tv/s/article/how-to-use-automod?language=en_US, 2021.
[60] G. Morrow, B. Swire-Thompson, J. M. Polny, M. Kopec, and J. P. Wihbey. The emerging science of content labeling: Contextualizing social media content moderation. *Journal of the Association for Information Science and Technology*, 73(10), 2022.
[61] S. Myers West. Censored, suspended, shadowbanned: User interpretations of content moderation on social media platforms. *New Media & Society*, 2018.
[62] V.-H. Nguyen, K. Sugiyama, P. Nakov, and M.-Y. Kan. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM international conference on information knowledge management*, 2020.
[63] P. Ogilvie and J. Callan. Experiments using the Lemur toolkit. In *TREC*, volume 1, 2001.
[64] D. Oleson, A. Sorokin, G. Laughlin, V. Hester, J. Le, and L. Biewald. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. In *Workshops at the Twenty-Fifth AAAI conference on artificial intelligence*, 2011.
[65] R. M. Palau and M.-F. Moens. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, 2009.
[66] J. Z. Pan, S. Pavlova, C. Li, N. Li, Y. Li, and J. Liu. Content based fake news detection using knowledge graphs. In *International semantic web conference*. Springer, 2018.
[67] O. Papakyriakopoulos and E. Goodman. The Impact of Twitter Labels on Misinformation Spread and User Engagement: Lessons from Trump's Election Tweets. In *The Web Conference (WWW)*, 2022.
[68] Y. Park, D. McCoy, and E. Shi. Understanding craigslist rental scams. In *International Conference on Financial Cryptography and Data Security*, 2016.
[69] R. K. Pasumarthi, S. Bruch, X. Wang, C. Li, M. Bendersky, M. Najork, J. Pfeifer, N. Golbandi, R. Anil, and S. Wolf. Tf-ranking: Scalable tensorflow library for learning-to-rank. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2019.
[70] G. Pennycook, A. Bear, E. T. Collins, and D. G. Rand. The implied truth effect: Attaching warnings to a subset of fake news headlines increases perceived accuracy of headlines without warnings. *Management Science*, 66(11), 2020.
[71] G. Pennycook, T. D. Cannon, and D. G. Rand. Prior exposure increases perceived accuracy of fake news. *Journal of experimental psychology: general*, 147(12), 2018.
[72] F. Qian, C. Gong, K. Sharma, and Y. Liu. Neural User Response Generator: Fake News Detection with Collective User Intelligence. In *IJCAI*, volume 18, 2018.
[73] T. Qin and T.-Y. Liu. Introducing LETOR 4.0 datasets. *arXiv:1306.2597*, 2013.
[74] T. Qin, T.-Y. Liu, J. Xu, and H. Li. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4), 2010.
[75] J. Ramos et al. Using TF-IDF to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.
[76] R. Recabarren, B. Carbunar, N. Hernandez, and A. A. Shafin. Strategies and Vulnerabilities of Participants in Venezuelan Influence Operations. *arXiv:2210.11673*, 2022.
[77] N. Reimers and I. Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019.
[78] S. Robertson, H. Zaragoza, et al. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4), 2009.
[79] S. Rose, D. Engel, N. Cramer, and W. Cowley. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1(1-20), 2010.
[80] B. Ross, A. Jung, J. Heisel, and S. Stieglitz. Fake news on social media: The (in) effectiveness of warning messages. In *International Conference on Information Systems*, 2018.
[81] M. H. Saeed, S. Ali, J. Blackburn, E. De Cristofaro, S. Zannettou, and G. Stringhini. TROLLMAGNIFIER: Detecting State-Sponsored Troll Accounts on Reddit. In *IEEE Symposium on Security and Privacy*, 2022.
[82] J. Seering. Reconsidering community self-moderation: the role of research in supporting community-based models for online content moderation. *CHI Conference on Human Factors in Computing Systems*, 4, 2020.
[83] J. Seering, T. Wang, J. Yoon, and G. Kaufman. Moderator engagement and community development in the age of algorithms. *New Media & Society*, 21(7), 2019.
[84] C. Shao, G. L. Ciampaglia, A. Flammini, and F. Menczer. Hoaxy: A platform for tracking online misinformation. In *The Web Conference (WWW)*, 2016.
[85] C. Shao, P.-M. Hui, L. Wang, X. Jiang, A. Flammini, F. Menczer, and G. L. Ciampaglia. Anatomy of an online misinformation network. *Plos one*, 2018.
[86] F. Sharevski, R. Alsaadi, P. Jachim, and E. Pieroni. Misinformation Warning Labels: Twitter's Soft Moderation Effects on COVID-19 Vaccine Belief Echoes. *arXiv:2104.00779*, 2021.

[87] B. Shaw, J. Shea, S. Sinha, and A. Hogue. Learning to rank for spatiotemporal search. In *ACM international conference on Web search and data mining (WSDM)*, 2013.

[88] K. Shu, X. Zhou, S. Wang, R. Zafarani, and H. Liu. The role of user profiles for fake news detection. In *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, 2019.

[89] L. Singh, S. Bansal, L. Bode, C. Budak, G. Chi, K. Kawintiranon, C. Padden, R. Vanarsdall, E. Vraga, and Y. Wang. A first look at COVID-19 information and misinformation sharing on Twitter. *arXiv:2003.13907*, 2020.

[90] J. Smith, G. Jackson, and S. Raj. Designing against misinformation. https://medium.com/designatmeta/designing-against-misinformation-e5846b3aa1e2, 2017.

[91] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33, 2020.

[92] G. Stanovsky, J. Michael, L. Zettlemoyer, and I. Dagan. Supervised open information extraction. 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.

[93] K. Starbird, A. Arif, and T. Wilson. Disinformation as collaborative work: Surfacing the participatory nature of strategic information operations. *ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW)*, 2019.

[94] G. Suarez-Tangil, M. Edwards, C. Peersman, G. Stringhini, A. Rashid, and M. Whitty. Automatically dismantling online dating fraud. *IEEE Transactions on Information Forensics and Security*, 2019.

[95] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers on large online QA collections. In *proceedings of ACL-08: HLT*, 2008.

[96] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers to non-factoid questions from web collections. *Computational linguistics*, 37(2), 2011.

[97] F. Tahmasbi, L. Schild, C. Ling, J. Blackburn, G. Stringhini, Y. Zhang, and S. Zannettou. "Go eat a bat, Chang!": On the Emergence of Sinophobic Behavior on Web Communities in the Face of COVID-19. In *The Web Conference (WWW)*, 2021.

[98] S. Tschiatschek, A. Singla, M. Gomez Rodriguez, A. Merchant, and A. Krause. Fake news detection in social networks via crowd signals. In *The Web Conference (WWW)*, 2018.

[99] Twitter. An update on our work around the 2020 US Elections. https://blog.twitter.com/en_us/topics/company/2020/2020-election-update, 2020.

[100] Twitter. The Twitter Rules. https://help.twitter.com/en/rules-and-policies/twitter-rules, 2020.

[101] Twitter. Our approach to the 2022 US midterms. https://blog.twitter.com/en_us/topics/company/2022/-our-approach-to-the-2022-us-midterms, 2022.

[102] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 2018.

[103] A. Wührl and R. Klinger. Claim Detection in Biomedical Twitter Posts. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, 2021.

[104] S. Zannettou. I Won the Election: An Empirical Analysis of Soft Moderation Interventions on Twitter. In *AAAI International Conference on Web and Social Media (ICWSM)*, volume 15, 2021.

[105] S. Zannettou, T. Caulfield, W. Setzer, M. Sirivianos, G. Stringhini, and J. Blackburn. Who let the trolls out? towards understanding state-sponsored trolls. In *ACM Conference on Web Science (WebSci)*, 2019.

[106] S. Zannettou, J. Finkelstein, B. Bradlyn, and J. Blackburn. A quantitative approach to understanding online antisemitism. In *AAAI International Conference on Web and Social Media (ICWSM)*, 2020.

[107] D. Y. Zhang, D. Wang, and Y. Zhang. Constraint-aware dynamic truth discovery in big data social media sensing. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017.

[108] P. Zhang, A. Oest, H. Cho, Z. Sun, R. Johnson, B. Wardman, S. Sarker, A. Kapravelos, T. Bao, R. Wang, et al. Crawlphish: Large-scale analysis of client-side cloaking techniques in phishing. In *IEEE Symposium on Security and Privacy*, 2021.

[109] W. Zhu, H. Gong, R. Bansal, Z. Weinberg, N. Christin, G. Fanti, and S. Bhat. Self-supervised euphemism detection and identification for content moderation. In *IEEE Symposium on Security and Privacy*, 2021.