Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

DATA SCIENCE AND COMPUTATION

Ciclo 36

Settore Concorsuale: 09/H1 - SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

Settore Scientifico Disciplinare: ING-INF/05 - SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

ANALYSIS AND APPLICATION OF LANGUAGE MODELS TO HUMAN-GENERATED TEXTUAL CONTENT

Presentata da: Marco Di Giovanni

Coordinatore Dottorato

Andrea Cavalli

Supervisore

Marco Brambilla

Co-supervisore

Andrea Cavalli

Esame finale anno 2022

***Adrian Veidt**: I did the right thing, didn’t I? It all worked out in the end.*
***Dr. Manhattan**: "In the end"? Nothing ends, Adrian. Nothing ever ends.*

- Alan Moore, Watchmen

Abstract

SOCIAL NETWORKS are enormous sources of human-generated content. Users continuously create information, useful but hard to detect, extract, and categorize. Language Models (LMs) have always been among the most useful and used approaches to process textual data. Firstly designed as simple unigram models, they improved through the years until the recent release of BERT, a pre-trained Transformer-based model reaching state-of-the-art performances in many heterogeneous benchmark tasks, such as text classification and tagging. In this thesis, I apply LMs to textual content publicly shared on social media. I selected Twitter as the principal source of data for the performed experiments since its users mainly share short and noisy texts. My goal is to build models that generate meaningful representations of users encoding their syntactic and semantic features. Once appropriate embeddings are defined, I compute similarities between users to perform higher-level analyses. Tested tasks include the extraction of emerging knowledge, represented by users similar to a given set of well-known accounts, controversy detection, obtaining controversy scores for topics discussed online, community detection and characterization, clustering similar users and detecting outliers, and stance classification of users and tweets (e.g., political inclination, COVID-19 vaccines position). The obtained results suggest that publicly available data contains delicate information about users, and Language Models can now extract it, threatening users' privacy.

Contents

List of Figures

CHAPTER 1

Introduction and Summary

1.1 Social Networking Sites

Social networking sites are online platforms used by people to build social relationships [43]. In the last decade of the century, the first platforms became popular, but recently, their popularity exponentially increased, reaching in 2020 almost four billion active users [311][1][2][3][4][5]. The enormous success of social networking sites is partially due to the ease to sign in, since it usually requires a few minutes, and to post content restricted only by the Term of Service of the platform, and to see the content posted by other users in the social network, usually *friends* or famous people. Algorithms that control the daily feeds of users maximize the time spent by users on the platform. The selection of the most appropriate content is crucial.

The possibility to share texts, news, pictures, videos and songs, combined with the constant interaction with other users, results in large quantities of data constantly uploaded, downloaded, collected and stored. These data are valuable since social network

[1] https://s22.q4cdn.com/826641620/files/doc_financials/2020/q2/Q2-2020-Shareholder-Letter.pdf

[2] https://wearesocial.com/blog/2020/07/more-than-half-of-the-people-on-earth-now-use-social-media

[3] https://blog.linkedin.com/2015/10/29/400-million-members

[4] https://www.cnbc.com/2017/09/25/how-many-users-does-instagram-have-now-800-million.html

[5] https://www.wsj.com/articles/reddit-claims-52-million-daily-users-revealing-a-key-figure-for-social-media-platforms-11606822200

data scientists and researchers can exploit them to perform numerous heterogeneous tasks and analyses to obtain insights about the users and the platform itself.

There are many **social implications** of this phenomenon. Ethical issues [36, 166], privacy issues [25] and crowd control through targeted banners [106][6] are just some of the concerns of Social Networking sites. Even if investigations and discussions about implications are not the scopes of this thesis, everybody should be conscious of which data the social networks collect and share and how they use them. For example, automatic user profiling is a branch of artificial intelligence aiming to collect information about users to build *profiles*. Such profiles are essential for recommendation tasks [79, 108, 143, 227, 265], social threat detection [224] or even government-related tasks[7]. The possibilities are numerous and intriguing, but the results could be potentially dangerous. Thus many research conferences nominated Ethic Advisory Committees, and they recommend or impose ethical statements in published researches[8].

In this thesis, I study **human-generated data** from social networks. Usually, Social Networking Sites develop and release official APIs for easy but limited[9] access to their continuously generated datasets. The APIs control the downloaded data and prevent the illegal scraping of data. Given a query, the APIs return sets of data that are later stored locally. If the magnitude of requested data is too high, frequent when dealing with human-generated data from social networks, the API splits them into batches and returns them iteratively. This procedure allows control of which and how much data are shared by the social network. Thus, real-time applications should be aware of these limits. The speed of these algorithms is crucial and depends on the speed of the data collection. A classic example of real-time analyses on social networks is the detection of new trends. It requires up-to-date datasets since trends have a limited lifespan, and virality and reactivity play a crucial role in the process [147, 150].

The selection of the most appropriate **query** is also essential to obtain unbiased data sets. The noise of the data generated in social networks makes the collection extremely sensitive to the query, and often similar inputs result in different datasets that could bias further analyses. Works have been declared inaccurate because based on biased datasets due to wrong or incomplete data collection queries (e.g., collecting posts about an online debate by searching for texts shared with a selected hashtag mainly used by just one side of the controversy will result in an incomplete dataset) [126, 289]. Usually, researchers iteratively improve the final query by inspecting data of previous partial tests and the plausibility of the final results. However, this procedure is not always possible, especially when the data collection process is slow or when dealing

[6] https://en.wikipedia.org/wiki/Russian_interference_in_the_2016_United_States_elections

[7] https://en.wikipedia.org/wiki/Social_Credit_System

[8] https://2021.aclweb.org/ethics/Ethics-FAQ/

[9] https://developer.twitter.com/en/docs/twitter-api/v1/rate-limits

with real-time applications.

Finally, being often human-generated (if bots are not involved), shared content is usually **noisy, incomplete and unreliable** [279]. A careful data pre-processing and cleaning step is essential to perform accurate analyses [19, 52, 281]. Social Media users often use emojis, and the best way to process them is not straightforward: they can be ignored, used as single characters or even translated to corresponding texts. Shared URLs contain information that could be useful for many applications, such as the URL domain. Finally, typos and slangs are rare in supervised professional content but common in user-generated textual content, and models trained on formal data are usually not robust enough to process them correctly.

Different social networking sites usually focus on different types of content shared. Picking the right social network is essential to perform complete and meaningful analyses. For example, **Twitter**[10] is known and used to share tweets, short texts of 140 characters, recently increased to 280, **Instagram**[11] for pictures, **Youtube**[12] for long videos, **TikTok**[13] for short ones with music in the background. Moreover, not every social network is widely used everywhere [61], and some countries forbid citizens to use them[14]. When performing social network analysis, researchers and readers must be aware that there is always some bias included since active users of social networks are rarely homogeneously distributed through the whole population of the world or a selected country. Usually, there is an age distribution discrepancy: younger people and older people use different social networks, while the elderly are rarely active users. People with different interests prefer different platforms to interact, preferring specialized ones. Finally, some people do not interact with social networks at all.

Almost every work described in this thesis is performed on data from Twitter, being the most famous social network sharing mainly textual content. Its official API is well-documented, and the rate limits are large enough to perform the tasks described here. Even if it is not the most used social networking site in Italy, the volume of tweets and users collected in the experiments is big enough to perform statistically significant analyses (see Section 4.1, Section 5.2 and Section 6.2). Also, **Facebook**[15] is a used alternative to extract textual content, but the collected data has stricter limitations due to the different privacy policies (see Section 6.1).

This thesis is a collection of works in the field of social network analysis. **Social Network Analysis** is a research field where *social networks* are the main object of investigation. The first and most common approach is to look at the *graph* lying underneath the social network, usually built connecting users when they verify a spe-

[10] www.twitter.com

[11] www.instagram.com

[12] www.youtube.com

[13] www.tiktok.com

[14] https://en.wikipedia.org/wiki/Censorship_of_Facebook

[15] www.facebook.com

cific condition (e.g., they are “friends" or they “follow" each other). Many researchers study the structure of these multi-layer graphs and the mathematical laws that they follow [15,206,222]. Static and dynamic measures are defined to describe and differentiate the networks and their evolution. This branch of social network analysis is the oldest one, and the methods designed have been refined over the years, recently obtaining impressive results [16, 69, 83]. Some of the most relevant applications include: investigating communities of users by looking at how the users are connected [187, 229, 316] and what is their relative position in the graph and their centrality [259]; link prediction applied to suggest new interactions and interests (i.e. recommender systems) [148,189]; analyzing information flows in the graphs [37] like the spreading of real and fake news [235].

However, social networks are not *just* networks and users are not just linked together. Content is constantly generated, liked, commented and shared by users. Even if the social graph is not directly involved in the analyses, we can still consider as social network analysis the elaboration of the content produced in social networks.

Due to the earlier success of deep neural networks for image classification and object detection [173,250], **user-generated images** have been used earlier as inputs for studies on social networks, Facebook, Instagram and Flickr[16] being the principal sources [197, 287, 294]. One of the most urgent recent tasks is to detect inappropriate images that do not follow social network policies. Algorithms to automatically classify and report explicit pictures are essential since the task would have been impossible if humans had to check every shared image [26]. Interesting how tags helped to create automatic ground truth of images that are usually unlabeled. Users pick tags to describe shared images since it produces more views and likes. However, these tags can be used as noisy labels for classification, obtaining big datasets with low manual effort [62, 128]. Other applications of images shared on social networks include the detection and clustering of people faces, monuments and cities for automatic tagging or even prediction of the popularity of posts [169].

[16] https://www.flickr.com/

1.2 Textual Content and Natural Language Processing

User-generated textual content is also often shared in social networks alone or coupled with images or videos. The ease of posting textual content on social networks tempts users to continuously share ideas, thoughts, personal reflections about every trending topic. Sometimes the shared content contains hashtags, mentions to other users or links to other contents inside or outside the social network. Exploiting the magnitude and heterogeneity of these data is an intriguing and challenging task.

In this thesis, I deal entirely with textual content shared by users. The group of techniques developed to analyze natural language is called **Natural Language Processing** (NLP). NLP recently became extremely popular due to the performances of novel models, such as ELMo [231], BERT [91], RoBERTa [192], GPT-2 [241], GPT-3 [51] and T5 [242]. These models reached state-of-the-art performances due to the long and expensive pre-training procedures, designed as self-supervised learning, to obtain detailed representations of words and sentences, even in multi-lingual settings. Transfer Learning plays a crucial role in the training: models pre-trained on semi-supervised tasks are later supervisedly finetuned with task-specific datasets. Some of the most famous applications are Named Entity Recognition [282], Machine Translation [78], Question Answering [144], Summarization [145, 191], but many other challenging tasks are continuously conceived. Current research focuses on improving many aspects of LMs, such as better encoding of the order of tokens [305], processing longer texts [322], computational complexity reduction with linear attention [64] and interpretability of the models [34] and their biases [178].

The success of these models in benchmark tasks also opens new ways to analyze social networks from syntactic and semantic points of view, where the texts are not only considered as a collection of words (Bag-of-Words classical approaches), nor as a collection of hashtags or mentions (graph approaches) but as dense high-dimensional feature vectors encoding sophisticated textual proprieties. There are numerous applications of NLP and language models on social networks, ranging from classical sentiment analysis [201] to more challenging tasks such as sarcasm detection [164] and user profiling [172].

1.3 Summary

In this thesis, I report some selected works performed during my PhD. Each chapter contains sections with detailed descriptions of the research questions formulated, the experiments performed, and the results obtained. I performed minor changes from the original versions to adapt them to the context of the thesis. Every work reported here has been published to peer-reviewed international conferences or journals except Section 4.4, currently under review at TheWebConf2022. Please do not share it to maintain the anonymity of the work. The outline of the thesis is the following.

In Chapter 2, I introduce **Language Models**. I start exposing their definition and a list of classical sentence and word embeddings techniques (e.g., BoW, TF-IDF and Word2Vec [203]). These approaches are the first attempts to process natural language with algorithms. Then I report an overview of modern deep language models based on the attention mechanisms (Transformer [298], BERT [91]) and state-of-the-art variations (RoBERTa [192], StructBERT [305], Big Bird [322]). These models revolutionized the field of NLP, exploiting the concept of Transfer Learning since large corpora of unsupervised textual data are recently available to perform long pre-training steps. Later, pre-trained models are fine-tuned on specific tasks. I briefly summarize the models and their relative papers and emphasize the key ideas that lead to state-of-the-art results. This selection is not complete, but its scope is to introduce some of the principal successes in NLP. The most appropriate models have been selected and used as fundamental components in the following chapters.

In Chapter 3, I describe how to **extract knowledge** from social networks. I report the pipelines designed to extract selected users from social networks. Emerging users represent knowledge that is not already present in knowledge bases. These approaches are crucial when dealing with emerging knowledge, not yet formalized on ontologies of curated data but already circulating on social networks.

The first work describes a pipeline that, fed with a set of Twitter accounts (called seeds) belonging to the same community (e.g. Twitter accounts of fashion designers), outputs new Twitter accounts belonging to the same community. Syntactic and semantic features of the shared textual content are fundamental. Since the obtained results are accurate, I started to investigate how the performance of the previous pipeline changes when I iteratively use the outputs as new seeds of the algorithm, obtaining accuracies highly dependent on the community selected. To evaluate and rank the best candidates, we compute the similarity between feature vectors of users, computed with standard NLP approaches from the textual content. The second section of the chapter describes this iterative approach.

The obtained results suggest continuing the investigation in two directions: by selecting more challenging types of knowledge to extract and by designing more robust pipelines able to deal with noisy data.

I select the second direction for the works reported in Chapter 4, which contains studies about **communities detection, controversy detection and user semantic embeddings**. The main goal of these works is to compute accurate user embeddings. They are essential to perform further tasks such as knowledge extraction, the detection of similar users belonging to the same community, detection of controversies between communities and detection of outliers.

The first work, about the characterization of communities and classification of users, is conceived to understand and test the best way to compute similarities between users by looking solely at the textual content shared. This investigation is essential to improve the previously described knowledge extraction pipeline since user similarity is its fundamental component. Thus, a better approach to defining and computing similarities results in better extraction of knowledge. I successfully prove the hypothesis that users with common interests usually write and share semantically similar posts.

In the second work of the chapter, I study a common phenomenon in social networking sites: controversies. When two communities of similar users talk about the same topic from different points of view, there is a controversy. The work describes how to detect and quantify these controversies in online Twitter discussions using shared textual content. The proposed approach compares to the state-of-the-art graph-based techniques, where the retweet graph is used as the starting point to classify whether a discussion is controversial or not. However, I hypothesized that, even if the structure of the retweet graph reflects the sides of controversies, the content published should too. When users share opposite opinions about a topic, the content is different, and quantifying this difference with Language Models is crucial. This content-based approach achieves state-of-the-art results both in terms of accuracy and computational time on a dataset of 30 multilingual topics, half controversial and half not.

The third work investigates how Transformer-based models can improve the embeddings of tweets encoding semantic similarity. Since the previous studies apply classical approaches showing that a straightforward application of recent models is usually not enough to obtain better results, I train a model that outperforms previous techniques on Twitter-related tasks. The obtained embeddings are at the tweet level because the nature of deep language models does not allow longer inputs. The results prove that datasets obtained from social media are very useful when adapted to detect the semantic similarity of documents.

In the last section of the chapter, I describe a hierarchical approach to obtain accurate user embeddings, overcoming the length limitations proper of classical deep language models. The Stage-1 model is the same model trained in the previous section that successfully embeds single tweets into dense high-dimensional vectors, while the Stage-2 model merges single tweet embeddings into a final user embedding, using a small Transformer-based architecture. I finally check whether the obtained representations reflect our idea of similarity with visualizations of communities, outlier detection and polarization detection.

Chapter 5 contains two works where I investigated how Twitter users reacted to two **Italian political events**: the elections of March 2018 and the 2020 Constitutional Referendum. The first work describes the design of a political inclination classifier. Applying classical NLP approaches to the textual content of politicians, I was able to perform accurate predictions of their political party. However, the approach fails when instead of politicians, we try to analyze citizens. The main reason is that the political inclination of non-politician users is hard to collect as ground truth to evaluate the goodness of the trained models.

The second work describes a model that classifies the stance of tweets about the 2020 Italian constitutional referendum. The introduction of a hashtag-based semi-automatic ground truth allows the building of large training datasets used to train binary classification models. I evaluate the models on manually annotated data proving that they accurately predict the stance by solely looking at the textual content shared in tweets. I finally discuss the discrepancy between the recorded activity on Twitter and the Referendum outcome, emphasising possible biases affecting the dataset collection.

Chapter 6 describes three works that analyze social activities about **COVID-19 and vaccines against it**. Since the COVID-19 pandemic characterized the last years, I have decided to contribute to the research by investigating the textual content shared about this topic.

In the first work, I analyzed the “information disorders" during the first four months of the Facebook “infodemic" caused by COVID-19. While the other authors analyzed the network structure of users and the propagation of fake news (not included in this thesis), my contribution involves the linguistic analysis of Facebook posts. I computed the polarization of accounts, the distribution of embedding of texts posted by users sharing misinformation, and the correlation of the sentiment of posts during the selected time window with important events. The results prove that users sharing different types of misinformation are syntactically and semantically similar, different from users sharing only content from reliable sources.

The second work introduces Vaccinitaly, an ongoing project that monitors online conversations about COVID-19 vaccines in Italy. We built a platform that continuously collects data about vaccines from Twitter and Facebook, and we perform multiple analyses on the obtained posts and tweets, such as the correlation between the magnitude of misinformation shared and the vaccine acceptance, checking whether social networks influence the number of vaccinated citizens at a regional level.

The last work inspects in detail the textual content collected from Vaccinitaly. I trained a Transformer-based model to predict the stance of tweets about vaccines. The binary classifier obtains good results due to the application of adaptive fine-tuning, a pre-training technique that involves unsupervised data related to the topic. I plan to train a 3-classes classifier to obtain predictions of neutral tweets, but the hashtag-based semi-automatic approach cannot include this variation. Thus, alternative methods are required. The performances of the classifier suggest that a real-time implementation is feasible. We aim to accurately monitor Social Media posts to detect and forecast anomalies that reflect important events about the topic.

CHAPTER 2

Language Models Zoo

In this chapter, I define Language Models, and I describe some of the most famous ones. I start with simple classical approaches such as Bag-of-Words techniques and their variants, including n-grams and skip-grams models. Then I report some pre-trained alternatives that exploit machine learning techniques to learn useful representations of words. However, the success of deep models revolutionized the NLP field. Firstly LSTM followed by Attention have been stacked into large layers to build deep architectures with hundreds of millions of parameters. With sophisticated transfer learning techniques, these architectures were trained on large corpora of unlabeled data and finetuned on specific tasks to obtain state-of-the-art results. The success of BERT led to an increase of the interest from the NLP community to these approaches, proposing variants and upgrades. I conclude this chapter by describing alternatives that try to solve the main limitation of Transformer-based architecture: the intrinsic limit of the length of documents selected as inputs. I used many of the models described in this chapter in the other works in this thesis.

2.1 Language Model Definition

A language model (LM) is a probability distribution over sequences of words.

Given a ordered sequence of m words $w_1, w_2, ..., w_m$, a LM aims to compute the probability $P(w_1, w_2, ..., w_m)$ of the sequence. Following basic probability rules, we can rewrite this as

$$P(w_1, w_2, ..., w_m) = \prod_{i=1}^{m} P(w_i|w_1, ..., w_{i-1})$$

Ideally, a LMs assigns to each sequence of words a probability that represents how likely that sequence is. For example, a LM should assign a larger values to the sequence “The cat is on the table." than to the sequences “The cat is on the taable", “The cat are on the table" and “The elephant is on the table", since they contain respectively a typo, a grammatical error and an unlikely event.

However, the total number of ordered sequences N_s of words grows exponentially with respect to the total number of existing words $N_W = |W|$, following the general equation $N_s = N_W^m$. Computing every possible probability for long sequences ($m \gg 1$) and large dictionaries of words ($N_W \gg 1$ is unfeasible. WordNet contains more than 2×10^5 different words [113], thus the number of sequences of length 2 is about 4×10^{10}, and the number of sequences of length 10 is more than the estimated number of atoms in the world. Assumptions are essentials.

Firstly we assume that every word is independent on other words in the same document, assumption certainly not true. We obtain a simple model called **Unigram model**, where the probability of a sequence of words is the product of probabilities of single words.

$$P(w_1, ..., w_n) = \prod_{i} P(w_i)$$

Thus the sentence “The cat is on the table" can be modeled by the Unigram model as follows:

$$P(\text{“}the\text{”}, \text{“}cat\text{”}, \text{“}is\text{”}, \text{“}on\text{”}, \text{“}the\text{”}, \text{“}table\text{”}) = \\ P(\text{“}the\text{”})^2 P(\text{“}cat\text{”}) P(\text{“}is\text{”}) P(\text{“}on\text{”}) P(\text{“}table\text{”})$$

To use this model, we just need to estimate N_W probabilities: $P(w_i)$ for each word w_i in our dictionary. This assumption leads to a huge gain in computational complexity with respect to the general case described above, but, since the assumption is far from realistic, we do not expect accurate estimations.

The main weak point of the Unigram model is that the assumption that each word is treated as completely independent from its neighbors is too strong, far from being realistic. Moreover, the order of words is completely ignored.

To improve it, we can relax the previous assumption: a word depends only on the n previous words. We call the obtained model **n-gram model** and we can compute the required probability with the following equation.

$$P(w_1, ..., w_m) \simeq \prod_{i=1}^{m} P(w_i | w_{i-n+1}, ..., w_{i-1})$$

Unigram, Bigram, Trigram models are n-gram models with respectively $n = 1, 2, 3$.

We model the sentence “The cat is on the table" using a the Bigram model as follows:

$$\begin{aligned}&P(\text{“}\textit{the}\text{”}, \text{“}\textit{cat}\text{”}, \text{“}\textit{is}\text{”}, \text{“}\textit{on}\text{”}, \text{“}\textit{the}\text{”}, \text{“}\textit{table}\text{”}) = \\ &\quad P(\text{“}\textit{cat}\text{”}|\text{“}\textit{the}\text{”})P(\text{“}\textit{is}\text{”}|\text{“}\textit{cat}\text{”})P(\text{“}\textit{on}\text{”}|\text{“}\textit{is}\text{”})P(\text{“}\textit{the}\text{”}|\text{“}\textit{on}\text{”})P(\text{“}\textit{table}\text{”}|\text{“}\textit{on}\text{”})\end{aligned}$$

and using a Trigram model:

$$\begin{aligned}&P(\text{“}\textit{the}\text{”}, \text{“}\textit{cat}\text{”}, \text{“}\textit{is}\text{”}, \text{“}\textit{on}\text{”}, \text{“}\textit{the}\text{”}, \text{“}\textit{table}\text{”}) = \\ &\quad P(\text{“}\textit{is}\text{”}|\text{“}\textit{the}\text{”}, \text{“}\textit{cat}\text{”})P(\text{“}\textit{on}\text{”}|\text{“}\textit{cat}\text{”}, \text{“}\textit{is}\text{”}) \\ &\quad P(\text{“}\textit{the}\text{”}|\text{“}\textit{is}\text{”}, \text{“}\textit{on}\text{”})P(\text{“}\textit{table}\text{”}|\text{“}\textit{on}\text{”}, \text{“}\textit{the}\text{”})\end{aligned}$$

This class of models requires the estimation of N_W^n probabilities, one for each sequence of n words of the dictionary. The value of the hyper-parameter n can be adapted to the task, higher values corresponding to more accurate models, lower values for faster approaches. However, the estimation of probabilities of rare sequences of words requires huge amounts of data, and inaccurate values could bias the final results. Finally, longer range dependencies cannot be detected by these models and better models have been designed.

2.2 Classical approaches

In this section I describe count-based approaches, simple classical techniques to embed documents.

2.2.1 Bag-of-words model

Bag-of-words (BoW) model is a simple and commonly used apporach to represent documents as the bags of their words. Each document is firstly tokenized, the tokens are usually pre-processed, and then the occurrences of cleaned tokens generate the representation of the document.

For example, given the sentence (document):

“the old man and the sea"

we firstly tokenize it obtaining the list:

[“the", “old", “man", “and", “the", “sea"]

and then we apply a BoW model to obtain:

{“the":2, “old":1, “man":1, “and":1, “sea":1}.

A boolean variant that neglects multiplicity of tokens can be implemented obtaining instead:

{“the":1, “old":1, “man":1, “and":1, “sea":1}

The main weak point of BoW models is that the order of words is neglected, thus the document

“and man old sea the the"

has the same BoW representation as the document reported before.

Moreover, BoW does not deal with word sense disambiguation: words that have multiple meanings (e.g., “like") are considered the same word and occurrences are merged.

BoW model is often used to vectorize a document: given a string, BoW model outputs a numerical vector (BoW representation) that can be used for further tasks. For example, given the mapping “the" to the number 0, “old" to 1, “man" to 2, “and" to 3 and “sea" to 4, the BoW representation of

“the old man and the sea"

is the vector

[2, 1, 1, 1, 1, 0, 0, ..., 0]

with length equal to the number of tokens in the map and the value of zero for every position after the fifth. We call n_t the number of times the token t appears in the document.

One of the most common application of this vectorization technique is for document classification. Firstly a document is tokenized and vectorized with BoW approach and then it is used to train a machine learning approach. Documents with higher values on the same tokens will be classified in the same class. For example, if we want to classify the topic of a news article we will notice that a trained classifier will focus on selected sets of words, such as “politician", “senator", “government" for politics related news and “ball", “game", “soccer" for sport related news.

This approach usually is implemented neglecting stop words, a manually compiled list of words in the selected language that are so common that they could negatively bias the prediction of trained models. Out-of-vocabulary words are usually ignored or grouped together in a single element of the vector labelled as UNK.

BoW approach can vectorize documents of any length without specific adaptations.

2.2.2 Term Frequency (TF)

The term frequency model TF is a simple improvement of BoW model obtained normalizing the previous vector.

Thus, the example above becomes

[1/3, 1/6, 1/6, 1/6, 1/6]

This approach is useful when dealing with documents of different length, since a single word appearing once in a document of few words is considered more important than the same word appearing once in a long document.

2.2.3 TF-IDF

When dealing with multiple documents, TF-IDF (Term Frequency-Inverse Document Frequency) approach is a good alternative to a simple TF vectorization. It is the product of two statistics, the TF, already explained above (Section 2.2.2), and the Inverse Document Frequency:

$$TF - IDF(t, d, D) = TF(t, d)\dot{I}DF(t, D)$$

Given N as the total number of documents, $IDF(t, D) = log\frac{N}{1+|\{d \in D: t \in d\}|}$ where $|\{d \in D : t \in d\}|$ is the number of documents where the token t appears (the “1+" term is added to avoid division by 0). Thus, if the token “the" appears in all the documents $IDF(\text{“}the\text{”}, D) \simeq 0$, and so $TF - IDF(\text{“}the\text{”}, d, D) \simeq 0$ for any document d.

This approach usually can deal not only with stop words, but also with common words that are not categorized as stop words in general, but with high frequency in the specific topics. For example, when classifying general news, the token “game" could be useful to detect news about politics, but if our classification task is defined over articles about different sports, then the token “game" will lose importance being much more common in the whole dataset.

2.2.4 n-gram model

A simple tentative to include the order of tokens in the models is done using n-grams. An n-gram is a continuous sequence of n items (tokens) from a document. BoW model can be seen as an n-gram model with $n = 1$ (unigram). With $n = 2$ (bigram), the document above becomes

{“the old":1, “old man":1, “man and":1, “and the":1, “the sea":1}

While higher order models usually outperforms the unigram model, long-range dependencies are still neglected. In the example above, we connected the words “old" and “man" previously treated as independent, but between the word “man" and “sea" there is still no connection.

Usually including bigrams and higher order n-grams is crucial when the document includes name of entities with more than one token (i.e., “New York", “United States" are also treated as a single entity by these models).

2.2.5 Skip-gram model

Skip-grams are a generalization of n-grams models where tokens do not need to be consecutive to be connected, but may leave gaps. A k-skip n-gram is a length n subsequence of tokens distant at most k from each other. The example above of 1-skip bigrams includes both all the bigrams already listed above and the following ones

{“the man":1, “old and":1, “man the":1, “and sea":1}

2.2.6 Comments

The simplicity of these models is a useful advantage when the computational cost is crucial, but they have clear throwbacks.

They usually lead to sparse representations of documents, since often a document includes a small number of terms with relation to the total number of tokens (words) N_W or n-grams. For example, a sentence of 5 different words is represented as a vector of dimension about 10^5 (the number of English words in WordNet) with only 5 not-zero entries.

Moreover, similar tokens are not correlated to each other but every token is independent. Different tokens are represented as *perpendicular* directions, thus synonyms are not correlated to each other. Their relationship and distance are the same as any other couple of words.

Even if these techniques are fast and easy to implement, they usually do not lead to state-of-the-art results and advanced dense alternatives have been designed and tested.

2.3 Pre-trained word representation

2.3.1 Word2Vec

Word embeddings have been revolutionized by Word2Vec [203], a data-based approach to compute high quality vector representation of words. While previous approaches treat words as independent, where each token is mapped in a perpendicular space, neglecting the semantic proprieties and similarities between tokens (as described above), this method computes dense representations based on co-occurrences of words in large datasets (1.6 bilion words from Google News dataset). It uses neural networks to estimate continuous representations of words, with two different approaches (CBOW and Skip-gram) inspired by the seminal work of [23].

Continuous Bag-of-Words (CBOW) predicts a word $w(t)$ from the context words ($w(t-2)$, $w(t-1)$, $w(t+1)$ and $w(t+2)$). The input words are firstly encoded in a V-dimensional one-hot vector, since strings cannot be used as inputs of a neural network. Then, they are embedded using the same projection matrix $w \in R^{V \times D}$, where $D = 300$ is the dimension of the embeddings. The embeddings are averaged obtaining a single D-dimensional vector. Finally, a hidden layer followed by softmax (usually implemented as hierarchical softmax) computes an output probability vector z of dimension V, where z_i is the probability that the word $w(t)$ is the i-th word of the vocabulary.

Skip-gram, instead, predicts the context words from the target word $w(t)$ with a technique similar to CBOW.

These models are trained on a Google News corpus of about $6B$ tokens, with a restricted vocabulary of one million words, with Adagrad optimizer. The training is performed in an unsupervised way, randomly picking the words to predict, learning the embedding matrix w. When the model is trained, we inspect the matrix of weights w, as it represents a mapping between each word and a D dimensional vector.

The impressive result is that similar words are mapped to similar vectors, with interesting proprieties such as:

$$w(\text{“}Paris\text{”}) - w(\text{“}France\text{”}) + w(\text{“}Italy\text{”}) \sim w(\text{“}Rome\text{”})$$

$$w(\text{“}king\text{”}) - w(\text{“}man\text{”}) + w(\text{“}woman\text{”}) \sim w(\text{“}queen\text{”})$$

$$w(\text{“}bigger\text{”}) - w(\text{“}big\text{”}) + w(\text{“}cold\text{”}) \sim w(\text{“}colder\text{”})$$

Vectorized words embedded both syntactic and semantic information, such as Paris is to France what Rome is to Italy, the female version of king is queen, if the comparative of big is bigger, than the comparative of cold is colder.

However, even if this approach collected many state-of-the-art results, it still maps each word to a vector, thus the context of the word is not taken into consideration at embedding time, but only during the training approach. The same word will be always mapped to the same vector, once the model is trained. This model also does not solve word sense disambiguation: the word "like" obtains the same dense representation even if it can be used with multiple meanings.

2.3.2 FastText

FasText is a fast text classifier developed in 2018 [32]. As the name suggests, its main benefit is that it compares in terms of accuracy to other bigger and deeper models, being much faster in both training and evaluation phases.

The architecture is simple and can be summarized as follows. Firstly, the input tokens x_i (N n-gram features) are multiplied by a weight matrix A (look-up table of words). The representations are averaged into a unique text representation of fixed length (not dependent to the number of tokens), that is then fed to a linear classifier. The output is computed with softmax (or hierarchical softmax if the number of classes is big). It is trained minimizing the equation

$$softmax(BAx_n)$$

where Ax_n is the average of the Ax_i representations of x_i tokens, and B represents the weights of the linear classifier.

FastText model is similar to CBOW model (previously described in Section 2.3.1), but the prediction is now a label over the predefined classes and not a word over the vocabulary.

FastText is a simple fast baseline method for text classification, being able to process billions of tokens in a small amount of time. It obtained state-of-the-art results in simple tasks that does not require a high representational power, being FastText a shallow model.

2.3.3 GloVe

GloVe (Global Vectors) [230] is a global log-bilinear regression model for unsupervised learning of word representations. Being very similar to Word2Vec, its better performances are due to global co-occurrence counts employed instead of local context windows.

The authors propose a weighted least squares regression model described by the following equation:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w_j} + b_i + \tilde{b_j} - \log X_{ij})^2 \tag{2.1}$$

where X is the word-word co-occurrence matrix, $w \in \mathbb{R}^d$ are word vectors to learn, $\tilde{w} \in \mathbb{R}^d$ are separate context word vectors, b and $\tilde{b}$ are biases, and f is the weighting function:

$$f(x) = \begin{cases} (x/x_{max})^{\alpha} & if\ x < x_{max} \\ 1 & otherwise \end{cases} \tag{2.2}$$

The authors trained the model on corpora of different sizes; the larger includes 42 billion tokens of web data from Common Crawl, tokenized and lowercased. A clear correlation between corpus size and performance is observed for syntactic tasks but not for semantic ones.

GloVe outperforms other models, including both versions of word2vec, on tasks such as word analogies (a is to b as c is to _?), word similarity and Named Entity Recognition (NER).

2.4 Deep contextualized word representation

2.4.1 ELMo

Polysemy is tackled by ELMo (Embeddings from Language Models) model [231]. Contrary to previous works (such as Word2Vec, Section 2.3.1), the representation of each token is now a function of the entire input sentence. Thus, words with multiple meanings are embedded into different vectors since their *context* will be different. ELMo uses LSTM layers and Bidirectional LMs to compute context-aware token embeddings.

A brief description of LSTMs and Bidirectional LMs follows.

Long Short-Term Memory networks (LSTMs)

Recurrent neural networks (RNN) are a simple variant of neural networks designed to process sequential inputs such as time series, voice signals and strings. An input signal is sequentially processed by the model, that updates a hidden state vector memorizing information about the signal. The straightforward application of RNN has to deal with issues related to long-term dependencies and vanishing gradients.

Long Short-Term Memory networks (LSTM) are an improvement of classical Recurrent Neural Networks (RNN) designed to overcome these issues. An LSTM cell (Figure 2.1) computes the following equations:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

$$D_t = \tanh(W_C[h_{t-1}, x_t] + b_C)$$

$$C_t = f_t C_{t-1} + i_t D_t$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \tanh(C_t)$$

where x_t is the input at time step t, h_t is the respective output and C_t is the cell state that intuitively conserves information about previous inputs. f represents the forget gate layer, i represents the input gate layer and o represents the output gate layer, σ is the sigmoid function, W are weight matrices and b are biases for the three gates.

Alternatives to LSTM includes GRU layers, designed to tackle long term dependencies and vanishing gradient with just two gates instead of three. However, usually LSTM and GRU layers obtain similar performances.

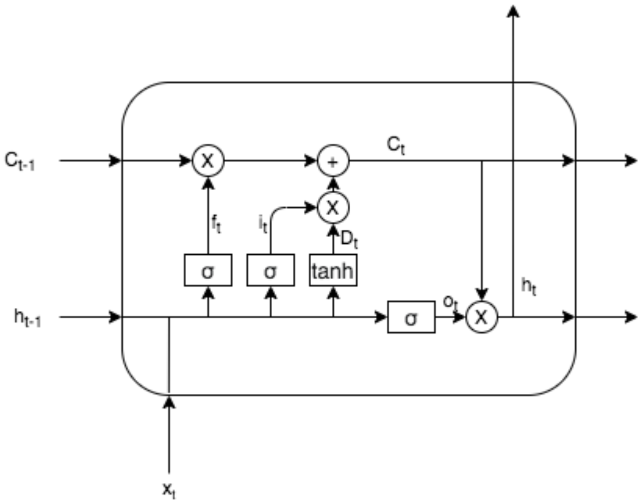


Figure 2.1: *LSTM architecture illustrated.*

Bidirectional LMs

Unidirectional LSTM, as the one described above, process the input sequentially starting from the beginning. They usually do not use information of the future to compute the present. This unidirectional approach is essential to specific tasks, but for NLP bidirectionality is crucial. Usually, words on a sentence depend also on following words, and sentences depend on following sentences. For example adjectives in English are often before the word they refer to while verbs in German are placed at the end of the sentence.

A forward Language Model computes the probability of a sentence by multiplying the probabilities of a token given its history:

$$P(t_1, ..., t_n) = \prod_{i=1}^{n} P(t_i | t_1, ..., t_{i-1})$$

Firstly, a context-independent token representation is computed and then fed to an L-layers forward LSTM. Top layer output is used to predict next token with softmax.

A backward LM is similar to forward LM, but instead of using the history of a token, it uses the future context:

$$P(t_1, ..., t_n) = \prod_{i=1}^{n} P(t_i | t_{i+1}, ..., t_n)$$

A biLM combines both a forward and backward LM, usually sharing only parameters of token representation and softmax layer. Including both previous context and future context is crucial to obtain state-of-the-art representations of tokens.

ELMo

ELMo combines intermediate layer representations of a biMLs. Each ELMO layer computes $2L + 1$ representations through LSTMs (L forward LSTMs, L backward LSTMs and the token representation). The representations are collapsed into a single vector ($ELMO_k$ for each token t_k). Finally, ELMo computes a task specific weighting of all BiLM layers obtaining:

$$ELMo_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j}^{task} s_j^{task} h_{k,j}^{LM}$$

where s_j^{task} are task specific softmax normalized weights and γ^{task} is a task specific scalar weight.

Given a pre-trained BiLM, ELMo collects all of the layer representations for each word, concatenates the ELMo vector with the context independent token representation and feeds it into a task RNN. It is observed that dropout and regularization of weights help the model to generalize better. The architecture is a pre-trained biLMs with $L = 2$ biLSTM layers of 4096 units and 512 dimension projections and residual connections. Context independent token representation is 2048 character n-gram convolutional filters, plus two highway layers and linear projection down to 512 dimension.

The model is evaluated on different standard NLP tasks, such as Question Answering, Textual entailment, Semantic role labeling, Coreference resolution, Named entity recognition and Sentiment analysis, outperforming state-of-the-art models. Ablation studies and controlled experiments confirmed the encoding capabilities of ELMo, storing efficiently syntactic and semantic proprieties of tokens. The authors observe that higher level LSTM states capture context dependent aspects of word meaning while lower level states model aspects of syntax.

2.4.2 Attention

An attention function is a mapping from a query and a set of key-value pairs to an output. The output is the weighted sum of the values, where the weight is computed by a compatibility function of the query with the key.

It was initially used to improve the performance of LSTM networks, obtaining the intuitive notion of attention when generating sequences of words. However, after the pubblication of “Attention is all you need" [298], architectures composed solely by Attention layers have been proposed obtaining a simpler and easier models to train, and resulting in state-of-the-art performances (Section 2.4.3) in many heterogeneous NLP tasks.

In the following sections, the mathematical formulation of attention is presented, followed by the description of the Trasformer, the first model solely composed by attention mechanisms.

Scaled Dot-Product Attention

The scaled dot-product attention is the basic operation of an attention layer. Given a query, a key and a value ($Q, K \in R^{b \times d_k}$ and $V \in R^{b \times d_v}$ respectively, where b is the batch size), attention is computed using the dot product of the query and the value, scaled properly with $\sqrt{d_k}$. Thus when the query and the value are similarly oriented, the dot result is higher than when they are oriented in opposite directions. The result is processed with a softmax function obtaining intuitively weights of how much attention must be used for each token in the sentence. Attention is the product of the softmax and the value matix of tokens.

Summarizing, the final equation shows as

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

where the factor $\frac{1}{\sqrt{d_k}}$ is for scaling purposes.

The query, key and value matrices are computed multiplying the input matrix $X \in R^{b \times m}$ with three different weight matrices: $W_Q \in R^{m \times d_k}$, $W_K \in R^{m \times d_k}$ and $W_V \in R^{m \times d_v}$, where m is the dimention of the raw input.

Multi-Head Attention

Instead of performing only a single attention function, researches prove that it is helpful to compute and merge many of them, using a multi-head attention mechanism. Firstly, different learned linear projections are computed to obtain multiple projected versions, focusing on different aspects of the inputs.

The general equation is the following

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

where

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

The linear projections matrices are $W_i^Q \in R^{d_{model} \times d_k}$, $W_i^K \in R^{d_{model} x d_k}$ and $W_i^V \in R^{d_{model} \times d_v}$, $W^O \in R^{hd_v \times d_{model}}$.

2.4.3 Transformer

The transformer [298] is the first architecture composed solely by attention mechanisms. It is designed following an encoder-decoder structure, where the encoder maps the inputs x to the sequence of continuous representations z, and the decoder maps z

to the output y, one at the time. The model is auto-regressive, using the previously generated tokens as inputs for the following predictions.

A brief description of the full architecture follows.

Encoder

The encoder is a stack of $N = 6$ identical layers, each composed of two sub-layers: a multi-head self-attention ($h = 8$ heads, $d_k = d_v = d_{model}/h = 64$) and a position-wise fully connected feed-forward network, with residual connections and layer normalization. The hidden dimension of the layers is $d_{model} = 512$.

Decoder

The decoder is a stack of $N = 6$ identical layers, like the encoder, plus a multi-head attention over the output of the encoder stack. Self-attention is modified to prevent positions from attending to subsequent positions, thus predictions can only depend on the known outputs at lower positions.

Finally, every layer in the encoder and in the decoder includes a fully-connected feed-forward network composed by two linear transformations with ReLU activation in between:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

where the inner dimension is $d_{ff} = 2048$.

Since, by nature, the attention mechanism treats tokens independently to their position in the sentence, this information should be encoded separately. The transformer uses positional encodings with sinusoidal functions:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$

and

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}})$$

The model obtained state of the art results on machine translation tasks (in particular, English-to-German and English-to-French translation tasks) outperforming models based on RNN. This suggests that attention mechanisms should not be applied to the outputs of RNN models, but architectures based solely on multi-head attention layers are better.

2.4.4 BERT

BERT (Bidirectional Encoder Representations from Transformers) [91], developed in 2018, is a language model that revolutionized NLP. It obtained state-of-the-art results

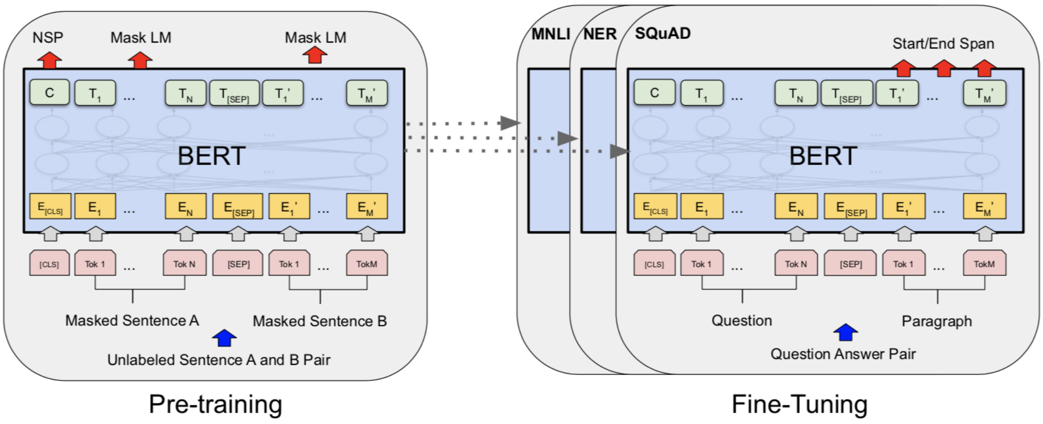


Figure 2.2: *BERT architecture illustrated. Image from [91] licensed under Creative Commons Attribution 4.0 International License.*

in many benchmarks such as GLUE [303] and SQuAD [245], proving that it reaches higher levels of language understanding (outperforming OpenAI GTP-1 [240]). This model was designed to be firstly pre-trained unsupervisedly and later finetuned with task-specific objectives (Figure 2.2).

One of the key ideas of BERT is to jointly condition on both left and right context in every layer to get representations from unlabeled text. Without substantial architecture changes from the Transformer previously described, its general usage was innovative and its public release was followed by a large number of papers about improvement, variants, analyses, interpretations and applications.

In the following sections, the architecture, the training procedure (pre-training and fine-tuning) and the results are described here.

Architecture

BERT architecture takes inspiration to the architecture of the Transformer: a multi-layer bidirectional Transformer encoder. It has been released initially in two versions: $BERT_{BASE}$, the base version with $L = 12$, $H = 768$ and $A = 12$; $BERT_{LARGE}$, the large version with $L = 24$, $H = 1024$ and $A = 16$, where L is the number of layers, H is the hidden size and A is the number of self-attention heads. Multiple size variants have been studied in [293], investigating the relation between number of parameters and performances.

Input/Output Representations

One of the main advantages of BERT is its task-independent design. Both single sentence and couple of sentences tasks can be addresses by BERT, since an unambiguous representation both single sentence and pair of sentences in one token sequence was designed. A sentence is tokenized with WordPiece, obtaining embeddings using a 30000 token vocabulary. The first token is always the [CLS] token, while the [SEP] token is used to separate sentences. Then, a sentence embedding is added to indicate whether a token belongs to the first or second sentence, and a positional embedding is added, similar to the one described for the Trasformer.

Using this approach both single sentence and pair of sentences tasks are addressed unambiguously by the same architecture.

Pre-training

BERT pre-training is performed in an unsupervised fashion on two tasks, Masked LM (MLM) and Next Sentence Prediction (NSP), on BooksCorpus (800M words) and English Wikipedia (2,500M words).

Masked LM (MLM) is performed by masking a percentage (15%) of input tokens at random. The hidden vectors, corresponding to the masked tokens ([MASK]), are fed into a softmax layer over the vocabulary. To leverage the mismatch between pre-training and finetuning (where the token [MASK] never appears), only 80% of the selected tokens are replaced with [MASK] token, 10% with a random token and the final 10% is unchanged. This task helps the model understand the relationship between a word and its neighbors.

Next Sentence Prediction (NSP) is the task of predicting whether, given two sentences A and B, B is the sentence that follows A. The task is performed by feeding the model with couples of sentences, 50% of them consecutive pairs and 50% where A and B are randomly picked. Also this task is unsupervised and helps the model understand the relationship between sentences and their neighbors.

Fine-tuning

After pretraining, the finetuning step is straightforward. It is a task-specific step performed in a supervised fashion using a selected dataset. The size of finetuning dataset is not crucial since the model already received a strong pre-training, thus, even applied to small datasets, BERT can reach performances much better than models solely trained on the final tasks.

The finetuning step is performed by adding a single feed-forward fully connected layer on the top of BERT. The inserved layer is task-dependent, thus it depends whether the goal is binary classification, multi-class classification or regression. The embedding of the [CLS] token is selected as the input features of the final layer. The full model is trained, without freezing its pre-trained weights.

The finetuning step is relatively inexpensive compared to pre-training both because the finetuning datasets are usually small, and because the model already learnt essential information during the pre-training phase.

2.4.5 Sentence Embedding Models

The approaches described above generate informative word embeddings. However, the need for fixed-length sentence embeddings to feed machine learning algorithms for many NLP tasks led researchers to investigate the best approaches to model sentences.

In this section, I describe some of the most used sentence-embedding approaches, and I focus on their strengths and weaknesses.

Classic approaches One of the straightforward sentence-embedding approaches is the previously described Bag of Word model. It generates fixed-size high-dimensional sparse embeddings that do not include word-order information.

Alternative standard techniques compute weighted averages of word-embeddings obtained from approaches such as Word2Vec and GloVe. These alternatives are more robust than BoW since they map similar words to similar vectors, but they also cannot exploit word-order information.

Doc2vec Doc2vec [185], also called Paragraph Vector, is an unsupervised approach that produces accurate sentence embeddings. These embeddings are inputs used to train accurate machine learning algorithms.

The algorithm generates two embedding matrices: W represents word embeddings, and D represents paragraph embeddings and acts as a memory of the current context.

The training is performed by window-sampling words whose vector representations are concatenated with a shared paragraph representation and fed to a classifier that predicts the following words. The model, trained with backpropagation and stochastic gradient descent, obtains state-of-the-art performances on sentiment analysis and information retrieval tasks.

Skip-Thought Skip-Thought [170] is an alternative to generate generic dense sentence vectors. The model has an encoder-decoder architecture, where the encoder is an RNN with GRU activations and the decoder is an RNN with conditional machine translation.

During training, the encoder is fed with sentences, and it generates an embedding used by two decoders. The goal of the decoders is to generate the sentences that were before and after the input sentence in the original corpus.

The authors prove that the obtained embeddings are informative and robust inputs to train linear classifiers and get state-of-the-art performances on many tasks such as semantic relatedness, paraphrase detection, image-sentence ranking, question-type classification and sentiment and subjectivity classification.

SIF Smoothed Inverted Frequency (SIF) [12] is a simpler approach that aims to modify the weighted average of word vectors using PCA/SVD to generate a strong baseline of unsupervised sentence embeddings. SIF models the probability that a word w is emitted in the sentence s with the equation:

$$P[w|c_s] = \alpha p(w) + 91 + \alpha)\frac{\exp(< \tilde{c}_s, v_w >)}{Z_{\tilde{c}_s}} \tag{2.3}$$

where c_s is the discourse vector and represents "what is being talked about", $\tilde{c}_s = \beta c_0 + (1-\beta)c_s$, $c_0 \perp c_s$, α and β are hyperparameters, $Z_{\tilde{c}_s}$ is the partition function and $p(w)$ is the unigram probability of word w.

The obtained sentence embeddings maximize the likelihood estimate for the vector c_s, which is approximately equal to the weighted average:

$$\arg\max \sum_{w \in s} \frac{a}{p(w) + a} v_w \tag{2.4}$$

where the weight $a = \frac{1-\alpha}{\alpha Z}$ is small for frequent words.

The authors evaluate the approach on 22 textual similarity datasets, and its embeddings are tested to be useful features for classification tasks, outperforming previous approaches.

SDAE Sequential Denoising AutoEncoder (SDAE) [152] is a sentence embedding model based on denoising autoencoders. The noise in sentences is generated by functions that delete words or swap pairs of words.

The architecture of SDAE is a classical encoder-decoder LSTM, trained to predict the original denoised source sentence from BookCorpus and obtains state-of-the-art performances on a wide range of supervised tasks.

FastSent The same authors of SDAE also proposed FastSent [152], a simple alternative to tackle especially unsupervised tasks.

This model exploits a type of sentence-level distributional hypothesis. Given consecutive sentences S_{i-1}, S_i, and S_{i+1}, where $s_i = \sum_{w \in S_{i-1} \cup S_{i+1}} u_w$ and u_w is a source embedding of word w, the cost is simply defined as

$$\sum_{w \in S_{i-1} \cup S_{i+1}} softmax(s_i, v_w) \tag{2.5}$$

where v_w is a target embedding of word w.

This model is much faster to train and obtains results comparable to the state-of-the-art on unsupervised datasets such as STS and SICK.

InferSent InferSent [72] is a deeper model supervisedly trained to generate universal sentence representations. The authors selected seven promising architectures to encode texts into vectors, such as LSTM, GRU, BiLSTM, Self-attentive networks and Hierarchical ConvNet.

They trained each model on the SNLI dataset (pairs of sentences labelled as entailed, neutral or contradictory) by feeding it with both input sentences one at a time. The obtained embeddings, u and v and concatenated with $u * v$ and $|u - v|$ and fed to a 3-class classifier consisting of multiple fully-connected layers and softmax.

Although trained on an NLI task, the embeddings outperform previous approaches on 12 transfer tasks, ranging from document classification to semantic relatedness and paraphrase detection.

USE Universal Sentence Encoder (USE) [58] includes two models trained both supervisedly on SNLI and unsupervisedly on Wikipedia, web news, question-answer pages and discussion forums, in a Skip-Though fashion.

The Transformer-based model outperforms the Deep Averaging Network (DNA) one due to its higher capacity, computational usage and memory cost.

The obtained embeddings obtain state-of-the-art results on many NLP transfer tasks, including sentiment analysis, subjectivity classification and semantic textual similarity.

2.4.6 Sentence BERT

Sentence BERT [252, 253] is an approach designed to compute semantic textual similarity (STS) of pairs of sentences. The authors propose this variant of BERT to decrease the computational time that the classical BERT model takes: they estimate that to find the most similar pairs of sentence in a collection of 10000 documents BERT requires 65 hours while Sentence BERT 5 seconds. The model has the same architecture of BERT but it is trained with a siamese or triplet approach [263]. The former receives as input two positive samples (i.e., semantically similar sentences) that are independently processed by a BERT-like model obtaining two embeddings. They are concatenated including also their difference and used as input for a softmax layer.

The triplet approach requires an anchor s_a, a positive sample s_p and a negative one s_n, and compute the following equation:

$$max(||s_a - s_p|| - ||s_a - s_n|| + \epsilon, 0)$$

with the margin $\epsilon = 1$.

The models have been trained using datasets of pairs of sentences manually labeled with scores between 0 and 5, indicating how much semantically similar the samples are: SNLI [40] and MultiNLI [312]. The obtained models reach state-of-the-art performances on STS benchmark [57] and have also been tested on other classification tasks from SentEval [71], obtaining results comparable to other approaches.

A multilingual approach was also proposed by the same authors, based on the idea that translated sentences should be mapped to the same location in the vector space as the original ones. They evaluate the model on more than 50 languages obtaining impressive results. They also investigated unsupervised approaches to overcome the need of paired data to perform the training of the models [304].

2.5 After BERT

2.5.1 RoBERTa

Robustly optimized BERT approach (RoBERTa) is an improvement of BERT [192]. The authors propose a new model with the same architecture as BERT, but investigating every design choice made and selecting the best alternatives. The resulting model outperforms BERT in many benchmark tasks suggesting that, even if the current trend is to build bigger models and bigger datasets [241], training details are essential to reach state-of-the-art performances.

The variations performed with relation to the original BERT model are summarized here.

- Instead of Static masking, used by BERT, RoBERTa uses dynamic masking when pre-trained on MLM. It generates the masking pattern every time a sentence is fed into the model, not during data preprocessing. This assures a masking always different, crucial when the pretraining is performed increasing the number of epochs and the size of datasets;

- Next Sentence Prediction (NSP) task is essential in BERT, since its removal hurts the performances of the model in many tasks. However, there are several alternatives training formats investigated with RoBERTa:

  1. SEGMENT-PAIR+NSP (BERT) includes the NSP loss, and each input is a pair of segments (since it can contain multiple sentences) with total combined length at most 512 tokens;
  2. SENTENCE-PAIR+NSP includes NSP loss, but each input is a pair of natural sentences, sampled from documents (significantly shorter than 512);
  3. FULL-SENTENCES does not include NSP loss, and each input is a set of full sentences sampled contiguously (can cross documents) with total length at most 512 tokens;
  4. DOC-SENTENCES is equal to FULL-SENTENCES but sentences cannot cross documents.

  The authors test every variant and prove that approaches like FULL-SENTENCES or DOC-SENTENCES improves the overall performance of the model. Probably when BERT was implemented without NSP and the authors found a decrease in performances, they picked SEGMENT-PAIR approach, not FULL-SENTENCES or DOC-SENTENCES variants, as performed in RoBERTa;

- The training is performed with larger batch size, up to $8K$, instead of 256 (BERT);

- Byte-Pair Encoding (BPE) is an alternative representation of strings in tokens that is considered between character-level representations and word-level representations, using subwords units, extracted by performing statistical analysis of a training corpus. It uses bytes instead of unicode characters as the base subwords units so that there is no need to introduce any *UNK* token for unknown strings. BERT uses BPE vocabulary of size $30K$ with heuristic tokenization rules. RoBERTa uses a larger vocabulary, with $50K$ subwords units without preprocessing or tokenization;

2.5.2 StructBERT

StructBERT [305] is a variation of BERT designed to incorporate language structures into pre-training. The authors state that BERT original pre-training objectives are not enough to get the underlying language structures, so they propose two auxiliary tasks:

- **Word Structural Objective**: jointly performed with the original masked LM objective, this task helps BERT to explicitly model the sequential order of words. Firstly, 15% of tokens are masked. Then, $K = 3$ tokens are shuffled and a softmax classifier on the top of the model is asked to predict the original order. The training is performed by maximizing the likelihood of placing every shuffled token in its real position.
- **Sentence Structural Objective**: instead of the original Next Sentence Prediction task, this task is designed to predict whether a sentence is the next sentence, the previous sentence or a random sentence. This adds bidirectional perception to the model.

The authors train architectures similar to BERT base and large, with comparable optimization techniques, obtaining state-of-the-art results in benchmark tasks such as GLUE and SQuAD.

2.5.3 Tackling longer documents

The full attention mechanism scales quadratically with respect to the sequence length n. This limits the application of deep learning models since they are unable process long sequences due to memory limits. Recently, novel attention variants replaced the classical full attention mechanism with linear attentions so to apply transformers to longer sequences of data.

Longformer

They key idea of Longformer [22] is to substitute the full attention mechanism with a combination of three attention patterns that scales linearly with the sequence length:

1. **Sliding window**: each token attends $\frac{1}{2}w$ tokens before and $\frac{1}{2}w$ tokens after it. The complexity becomes $O(nw)$, so to be efficient, the window size w should be small compared to n;
2. **Dilated sliding window**: similar to sliding window, but with gaps of size dilation d. Not all layers will use this pattern, but the dilation d will be dependent on the depth of the layers;
3. **Global attention**: special tokens, such as [CLS] token or tokens about answers in QA tasks, attend all the other tokens symmetrically. Since the number of this special tokens is very low with respect to n, this attention mechanism complexity is $O(n)$.

The authors test the longformer on Autoregressive left-to-right language modelling (estimating the probability distribution of a character/token given its previous characters/tokens), in particular text8 and enwik8 [198] datasets, where they reach new state-of-the-art results. They processed sentences of maximum length 32K, instead of 512 as the original BERT and RoBERTa models. They also test the model with a procedure similar to BERT pretraining. MLM task is selected, but sentences of length up to 4K tokens are processed. They start from RoBERTa weights and continue the pre-training with window size $w = 512$, obtaining a model with the same complexity as RoBERTa. The Longformer outperforms RoBERTa on many question answering, coreference resolution and document classification tasks, but the authors plan to apply it also for summarization tasks.

Big Bird

The authors of Big Bird [322] also replace the quadratic attention with a mix of attention mechanisms linear with respect to the sequence length: random attention, window attention and global attention. Random attention makes a token attend r random tokens instead of the full sequence of tokens, reducing the computational complexity from $O(n^2)$ to $O(rn)$. Window attention makes a token attend its w nearest tokens, obtaining a complexity of $O(wn)$, as in the Longformer. Global attention makes the g most important tokens attend to all the others, obtaining a complexity of $O(gn)$, as in the Longformer. BigBird is a combination of the three attentions.

This variant of attention is justified with theoretical proofs (it is a universal approximator of sequence to sequence functions and it is Turing complete) and tested experimentally on different tasks requiring long input sequences. After a pre-training starting from RoBERTa public weights, it reaches state-of-the-art results in Question answering tasks, document classification, summarization and also genomics experiments, such as Promoter region prediction and Chromatin-profile prediction, by treating the DNA as a sequence of base pairs.

Hierarchical Transformer

A different approach to deal with long texts is exposed in [226]. The authors segments long texts into smaller chunks that fits into classical BERT models. The outputs are propagated into a recurrent or transformer layer that generates the final prediction. The final model is a 2-Stages Hierarchical model, whose first stage processes single chunks and its second stage merges different chunks into a single output.

The chunks of long texts are obtained fixing a chunk size of 200 tokens and a shift of 50 tokens, thus consecutive chunks highly overlap.

They mainly test two versions of Hierarchical Transformers:

- Recurrence over BERT (**RoBERT**): the Stage-2 layer is a recurrent neural network, in particular a 100-dimensional LSTM followed by two fully connected layers with ReLU and softmax. Due to its nature, LSTM does not need positional embeddings;
- Transformer over BERT (**ToBERT**): the Stage-2 layer is similar to RoBERT but the LSTM layer is replaced by a small Transformer model (2 layers). The nature of Transformer layers requires positional embeddings when dealing with ordered tokens in a document. When dealing with ordered chunks, the authors experiment with similar positional embeddings and results suggest no clear improvements.

The authors train for 1 epoch with Adam optimizer only the weights of the Stage-2 models, freezing the ones of Stage-1 pre-trained BERT. The frozen parameters are obtained both from the original BERT model and from a pre-trained version of BERT on their datasets, the latter performing much better.

The models are finally tested on three binary classification or multi classification tasks and they notice that the higher improvements with respect to simple averages are obtained on datasets with higher fraction of long documents.

2.6 Autoregressive Language Models

In the whole thesis, I focused on Autoencoding Language Models, also called encoder-only models. These models are fed with textual data and generate embeddings of words or sentences. Thus, they are mainly suited for text understanding tasks, such as classification, stance detection and similarity quantification.

However, Autoregressive Language Models (i.e., decoder-only models) are also a valid alternative. Different from Autoencoding LMs, they continuously generate text given numerical inputs. Thus, they are suited for text generation tasks or few-shot learning.

In this section, I summarize three revolutionary Autoregressive LMs, namely GPT, GPT-2, and GPT-3, released by OpenAI. They set state-of-the-art results in zero-shot, one-shot or few-shot learning, introducing a novel approach to use Transformers for NLP.

Even if they are undoubtedly worth investigating in future research, the works reported in this thesis do not analyze or apply these models. There are three main reasons for this choice. First, the topics selected do not deal with text generation or few-shot learning, making Autoencoding LMs the most reasonable choice. Moreover, OpenAI initially refused to make a public release of GPT-2's source code or model parameters when announcing it due to possible malicious usage. They only released the best model months later, in November 2019. Up to now, OpenAI released GPT-3 only through an API with restricted access. Finally, the size of the models is often prohibitive and, even if smaller alternatives like DistilGPT2 are available, they introduce another source of error.

2.6.1 GPT

Generative Pre-Training (GPT) [240] is a pre-training technique successfully applied to Language Models to improve their language understanding skills. The authors exploit the transfer learning paradigm to train an autoregressive language model on a large dataset in a semi-supervised approach and fine-tune its weights to reach state-of-the-art performances on a wide range of NLP tasks, including Natural Language Inference (NLI), Question Answering (QA), sentence similarity and classification.

The authors pre-train the model on a large dataset of unique unpublished books called BookCorpus. The model maximizes the following autoregressive likelihood:

$$L_1(U) = \sum_i \log P(u_i | u_{i-k}, ..., u_{i-1}; \Theta) \tag{2.6}$$

where k is the size of the context window. The probability P is modelled with a 12-layer Transformer-based decoder, whose weights are represented by Θ.

During the fine-tuning step, the output of the model h_l^m, where l is the number of layers and m is the number of tokens in the input sentence, is fed into an additional linear output layer to predict the label y:

$$P(y|x^1, ..., x^m) = softmax(h_l^m W_y) \tag{2.7}$$

where W_y are additional learnable parameters.

To fine-tune the model on structured tasks, the authors designed task-specific input transformations (e.g., concatenation of premise and hypothesis with a delimiter in between for textual entailment tasks). This trick allows the model to accurately perform on many different tasks without substantial task-specific architecture changes.

The model reached state-of-the-art performances on some NLP tasks, and the authors also showed acceptable results on zero-shot evaluation, i.e., the pre-trained model performs tasks without fine-tuning its parameters.

Finally, ablation studies confirm the superiority of Transformer layers over LSTM.

2.6.2 GPT-2

GPT-2 [241] is a 1.5B parameter Transformer expecially designed to perform zero-shot learning. Its architecture is similar to GPT's one, with few modifications mainly regarding layer normalization and the total size of the model.

The model obtained state-of-the-art performances on many tasks without proper fine-tuning. The model was only pre-trained with a Language Model objective similar to equation 2.6 on a large unsupervised dataset: WebText. WebText is a dataset, expressly created by the authors that collect scraped web pages curated and filtered by humans, relying on *karma* scores from Reddit. The dataset contains 8 million documents for a total of 40 Gb of text.

The model uses a Byte-level BPE encoding to represent input, a middle ground between character and word level tokenizations. The vocabulary size is 50.257: 50.000 merges, 256 base vocabulary tokens, and one special token. This tokenization technique allows the model to tokenize and process every combination of characters.

The authors evaluated the model on eight datasets across domains and task types, observing clear improvements on small datasets or datasets in which long-term dependencies are crucial. Tested tasks include Children's Book Test (LM on different categories of words), LAMBADA (prediction of the final word of long documents), Winograd Schema Challenge (resolve ambiguities in texts), reading comprehension, summarization, translation and question answering.

Finally, a test based on 8-grams proves a small but consistent overlap between the pre-training dataset and the evaluation datasets, suggesting that memorization does not play a crucial role in the process.

The release of GPT-2, even if its zero-shot performance was still far from being usable in practical applications, suggests that training bigger Language Models on larger datasets brings better results since GPT-2 is still underfitting on WebText. Moreover, the exploitation of the language to provide a flexible way to specify inputs and outputs is decisive in the field of multitask learning on NLP.

The original paper ends with examples of articles generated by GPT-2 fed with an introduction written by humans. The almost perfect usage of English syntax and long-term correlations of the generated output stunned the research community when the article was published.

2.6.3 GPT-3

GPT-3 [51] is a simple extension of GPT-2. The architecture of the models is equal but bigger: GPT-3 has 175B parameters split into 96 layers. The model was trained using the same objective on a larger dataset (a mixture of filtered CommonCrawl, WebText2, Books1, Books2 and Wikipedia) of about 300 billion tokens. The authors strongly evaluate the contribution of scaling model capacity and dataset size showing empirically that their model did not reach a performance plateau yet.

The authors evaluated the model on few-shot, one-shot, and zero-shot settings, without fine-tuning the weights of the pre-trained model. The tasks were formulated in a setting similar to how humans communicate. For example, few-shot learning was implemented inserting before the input sentence a few examples of the required task.

The model was evaluated on tasks including language modelling, closed book question answering, translation, Winograd-style tasks, common sense reasoning, reading comprehension, SuperGLue, NLI, arithmetic, word scrambling and manipulation, SAT analogies, and news article generation. The model exhibits strong performances, often comparable with state-of-the-art approaches supervisedly trained on the tasks.

However, the authors also observed limitations of GPT-3, such as generating repetitive, not coherent and contradictory documents, not understanding common sense physics, and wrongly comparing words and sentences. Moreover, GPT-3, like every Autoregressive model, lacks bidirectionality, affecting its performance on tasks that require re-reading. Finally, its pre-training objective lacks the notion of what is most important to predict, and the final model lacks context about the world. The authors propose to tackle these limitations through human help, images, or reinforcement learning.

The authors conclude their analysis by investigating potential misuse of GPT-3 (e.g., generate misinformation, spam, fraudulent academic essays), biases (gender, race and religion) and energy usage. Results suggest that GPT-3 is still not reliable enough to be used by threat actors. Moreover, the many biases on the model reflect biases on the training data like other LMs. Finally, the energy-intensive training will be amortized over the model's lifetime since GPT-3 is surprisingly efficient once trained.

CHAPTER 3

Knowledge Extraction from Social Media

In this chapter, I describe how to extract knowledge from Social Media. The term **Knowledge** is hard to define quantitatively in this scenario. Oxford dictionary defines knowledge as the *information, understanding, and skills that you gain through education or experience*[1].

Every day in the world countless events happen that change what we know. Hundreds of thousands of people are born each day[2], marry each other, have children, get sick, die. Governments change, make new laws, declare wars, fall, companies are founded, sold, bankrupt, new technologies are invented, new laws of nature are discovered and tested. Given the magnitude of events, it is increasingly difficult to keep up as the world evolves.

To formalize knowledge and make it easily accessible massive technologies have been used recently to produce very large ontologies: DBpedia, YAGO, the Knowledge Graphs in Google and Facebook derive from structured or semi-structured curated data [186, 249, 274, 280]. However, information evolves at a much faster pace. It is not easy for ontologies to be updated. New entities continuously emerge, and existing ones change or become obsolete.

Moreover, large ontologies collect high-frequency data, the most popular items, while often neglecting low-frequency data (entities that belong to the so-called *long*

[1] https://www.oxfordlearnersdictionaries.com/definition/american_english/knowledge

[2] https://www.worldometers.info/

tail, i.e. the portion of the entity's distribution having few occurrences [196]). The reason is that low-frequency data are usually harder to collect and check, and they do not contain as much valuable information as the high-frequency ones.

However, emerging entities not collected due to their low popularity today could be important in the future. Examples are small companies, emerging brands and young players, still not relevant enough to be included in ontologies.

To discover knowledge and its evolution, we can take advantage of powerful and massive sources: the content produced on social media. One can conjecture that somewhere, within such a massive content, any entity (and its evolution) has left some traces. The main challenge is that such traces are often unclassified, dispersed, disorganized, uncertain, partial, possibly incorrect. Therefore, deriving information about entities from social content is highly challenging.

In this chapter, I report and summarize two works about knowledge extraction from social networks. Firstly, I report the seminal work of [47], describing a pipeline to extract Twitter accounts of a selected domain (Section 3.1). The authors designed a pipeline that receives as input a set of user accounts (seeds) and returns candidate users similar to the selected seeds. Thus, if we choose fashion designers accounts, we obtain emergent fashion designers with this approach. In Section 3.2, we applied the same approach iteratively to investigate how knowledge evolves in time and space by using the extracted candidates as new seeds. Thus, with the term knowledge, we refer to users that belong to a selected domain. We investigate emerging knowledge by focusing on how known are the obtained users.

The works in this chapter have inspired the following research, exposed in the rest of the thesis. Starting from the problem of knowledge extraction, I investigate the best approach to compute similarities between users and how to use them applied to contemporary events.

3.1 Extracting Emerging Knowledge from Social Media

In this Section I report a brief summary of the paper Extracting Emerging Knowledge from Social Media [47]. This is a crucial research to understand the how to extract knowledge from Social Media, and how we designed the work in the next Section (Section 3.2).

Even if very large ontologies are constantly updated, knowledge evolves at a faster pace, making these ontologies incomplete, especially of low-frequency data. Usually ontologies collect high-frequency knowledge, stable and safe information, while low-frequency data are harder to find, collect and check. Socially produced content is a huge source of low-frequency data, being made by users. However, it is not completely reliable.

In this work, the authors design and test a method of discovering emerging entities from social networks. The emerging knowledge is represented by users belonging to a single field. The approach is quickly initialized by an expert of the selected field by selecting a small set of users (called seeds). A set of candidates is automatically extracted and, for each one of them, a feature vector is computed encoding the useful information about the user. Candidates are finally ranked using their distance from the feature vector of the seeds. The authors search between many syntactic and semantic variants to find the best combination of them that generates accurate user embedding, thus appropriate knowledge extraction.

Datasets

The approach is evaluated to three different domains:

- Fashion designers: 200 emerging brands in the Italian marked are collected by experts, with their twitter account, and used as seeds. 237000 tweets have been analyzed;
- Fiction writers: from a set of 100 writers engaged in a literature event in Australia, 22 seeds are collected and 14590 tweets have been analyzed;
- Live events: Universal Exposition (EXPO 2015) took place in Milan. 15 official accounts of exhibition pavilions are used as seeds, collecting 24000 tweets.

Methodology

The approach is initialized with a selection of seeds made by experts. They collect the social content posted by seeds and compute a feature vector for each one of them. To improve the overall approach, the authors introduce an outlier detection step to check whether seeds are homogeneous. They run PCA and k-means clustering ($k = 2$) (Coefficient Variation method), so that outliers (elements distant more than two standard deviations from the mean that could hurt the final performance) are detected and removed. Then the centroid of seeds is obtained as a global representative of the field, encoding the average features of the seeds.

The selection of candidates is performed merging the twitter handles in the social content posted by every seed. They hypothesize that seeds mention users related to them. Of course, not every mentioned user belongs to the same field of the seeds, thus they rank the obtained candidates by closeness to the centroid of seeds. However, the number of candidates obtained with this approach is too large to collect and process data for each one of them. The authors restrict the set of candidates using a ranking function with a selected threshold. This function prefers candidates mentioned by more seeds to candidates that, even if the total number of mentions is the same, are only mentioned by few seeds. Thus, the final score S_i of candidate i is computed using the following equation

$$S_i = \frac{a_i b_i}{(N - a_i + 1)}$$

where a_i is the number of seeds mentioning the $i - th$ candidate, b_i is the number of times a candidate is mentioned in the whole content and N is the number of seeds. If every seed mention the candidate N times once, its score is $S_i = N^2$, while is just one seed mention the candidate, its score is $S_i = 1$.

Candidates, ranked by score, are pruned so that only the content of the top ones is retrieved and processed to obtain a feature vector. Those vectors are successively compared to the centroid using cosine distance, Euclidean distance or Pearson correlation, measuring the similarity between those candidate users and the seeds selected by experts.

Feature vector computation

To compute the feature vectors, the authors test both syntactic and semantic approaches.

Syntactic methods are based on parts of Twitter texts explicitly defined as relevant. Two strategies are tested including using frequencies of all the handles and frequencies of all the handles and hashtags.

Semantic methods are based on a general knowledge base (DBpedia [29]). The expert initially selects also few DBpedia types relevant to the domain of the seeds. Dandelion[3], a commercial software, matches the texts with the pertinent semantic entities.

The strategies tested are the following (summarize in Table 3.1):

- AHE (All Handle and hashtag Entities): hashtags and handles that correspond to an entity in the knowledge base are selected;

[3] The free version of the API can be tested at https://dandelion.eu/

- CHE (Concrete Handle and hashtag Entities): hashtags and handles that correspond to an entity in the knowledge base, whose type is a concrete (i.e. most specialized) type, are selected;
- EHE (Expert Handle and hashtag Entities): hashtags and handles that correspond to an entity in the knowledge base, whose type is a selected type by experts, are selected;
- AHT (All Handle and hashtag Types): hashtags and handles that correspond to a type are selected;
- CHT (Concrete Handle and hashtag Types): hashtags and handles that correspond to a concrete (i.e. most specialized) type are selected;
- EHT (Expert Handle and hashtag Types): hashtags and handles that correspond to a type selected by experts are selected;
- AST (All Spot Types): any spot that correspond to a type are selected;
- CST (Concrete Spot Types): any spot that correspond to a concrete (i.e. most specialized) type are selected;
- EST (Expert Spot Types): any spot that correspond to a type selected by experts are selected;

Table 3.1: *Summary of strategies*

	HE	HT	ST
	(hastag and handle entities)	(hashtag and handle types)	(spot types)
A (all types)	AHE	AHT	AST
C (concrete types)	CHE	CHT	CST
E (expert types)	EHE	EHT	EST

Mixed strategies of HE and HT or ST have also been tested, defining $\alpha \in [0, 1]$ the mixing parameter, obtaining a total of 990 semantic strategy variants: 18 feature vector configurations (mixing one of the three HE strategies with one of the six HT or ST strategies), 11 values of α with a step of 0.1 and five levels of recall for the entity extraction algorithm: $[0.15, 0.25, 0.50, 0.75, 1]$.

Evaluation and conclusion

The evaluation metric is based on the usage scenario, since usually a small set of candidates is proposed to be evaluated by a user. They selected $precision@K$ as metric with $K = 10$, meaning that only the first 10 results (top 10 candidates similar to the centroid of seeds) are selected as output and evaluated. The evaluation is performed checking whether the output candidates are accounts related to the initialy selected topic (i.e.

with the Fashion designer dataset, if the candidates are actually accounts of fashion designers).

Results show that the best method is EHEˆAST with $\alpha = 0.7$ and $recall = 1$, obtaining acceptable precision scores. Lists of the best candidates are finally presented to experts to confirm their belonging.

The authors claim that their approach to extract emerging knowledge from social networks can be considered a first step on a larger project on knowledge discovery. The general pipeline has some limitations: the bias from experts providing seeds, the choice of Twitter as source being not appropriate for every field and the reliance on DBpedia that could not contain every concept needed.

3.2 Iterative Knowledge Extraction from Social Networks[1]

3.2.1 Introduction

In this section **SKE (the Social Knowledge Extractor)**, a method for discovering knowledge by extracting it from social content, is presented and tested in an iteratively fashion. The method is the evolution and extension of the research presented at WWW 2017 [47] (section 3.1) and is defined in the context of a broader vision on knowledge discovery (whose general framework is illustrated in [44]). We use **Twitter** as social content source; Twitter can be accessed via its public APIs, which extract tweets related to a given hashtag or Twitter account. We refer to **DBpedia** as generic source of ontological knowledge; DBpedia is publicly available through its open API. DBpedia *types* are used to partition the existing ontological knowledge, organized within a type hierarchy.

The domain of interest is described by a selection of DBpedia types. This selection is performed by domain experts and typically includes few (from five to ten) types. We find entities within such domain, by extracting them from the social content. The expert must also provide **seeds**, i.e. prototypes of the interesting entities, simply described by providing their twitter accounts. A small set of seeds is sufficient: we normally use 10 to 20 seeds at each call of SKE.

Once initialized by domain experts, the method is capable of finding entities by means of a mix of syntactic and semantic techniques. Our method collects information from the seed's tweets and generates **candidates**, i.e. other twitter accounts which are mentioned within the extracted tweets; then, it associates each candidate to a **feature vector**, built by using terms occurring in their social content, giving more relevance to terms which match the types selected by the expert; then it associates each candidate to a **score**, equivalent to the distance of each candidate from the centroid of the seeds; finally, it returns the top candidates, listed in decreasing score order. Once the candidates are generated, they can be forwarded to a **crowd of evaluators** that can assess the correctness of the extraction. Furthermore, the user can select a subset of candidates and reuse them as new seeds in a new execution of the knowledge extraction process.

In this section we study how the method captures *evolving knowledge*; this is a crucial aspect, as the method can be repeatedly applied in an iterative manner over the social content to capture new trends or to track knowledge spreading and evolution.

We answer the following research questions:

[1] Authors: Marco Brambilla, conceptualization, writing, editing; Stefano Ceri, conceptualization, writing, editing; Florian Daniel, conceptualization, writing, editing; Marco Di Giovanni, implementation, experiment design, writing; Andrea Mauri, conceptualization; Giorgia Ramponi, implementation, experiment design, writing. [45] WWW '18 Companion, April 23–27, 2018, Lyon, France ©2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY 4.0 License.

RQ1: *How does reconstructed domain knowledge evolve if the candidates of one extraction are recursively used as seeds?*

RQ2: *How does the reconstructed domain knowledge spread geographically?*

RQ3: *Can the method be used to inspect the past, present, and future of knowledge?*

RQ4: *Can the method be used to find emerging knowledge?*

This section is organized as follows: Section 3.2.2 describes our iterative knowledge extraction process; Section 3.2.3 presents the seven domains of interest over which we experiment the approach; Section 3.2.4 describes the four usage scenarios that respond to the above research questions and show the method at work on each of them; Section 3.2.5 describes our implementation; Section 3.2.6 discusses the related work; and Section 3.2.7 concludes.

3.2.2 Iterative Extraction Process

The **knowledge extraction process** we propose is briefly reported in Figure 3.1 and already described in the previous section:

1. *The user submits a set of seeds* in input as samples of concepts to search for. These seeds consist of Twitter handles (usernames);

2. *The user submits a set of expert types* as descriptors of the domain of interest;

3. *The extraction of new candidates* is then launched and proceeds as follows:

   (a) Elimination from the seeds of outliers according to principal component analysis and computation of the centroid of the filtered seeds;

   (b) Collection of all the posts of each seed;

   (c) Definition of the set of candidate new entities as all the user handles that are mentioned by the seeds (which may lead to several thousand candidates);

   (d) Filter of candidates based on *tf-df* similarity [47], which allows one to reduce the space of analysis of the candidates to a limited set of relevant ones;

   (e) Collection of all the posts of each candidate;

   (f) Computation of the feature vector representing each candidate;

   (g) Rank of the candidates based on the vectorial distance from the seed centroid and production of the result based upon the ranking.

4. Once the candidates are retrieved and ranked, the user can:

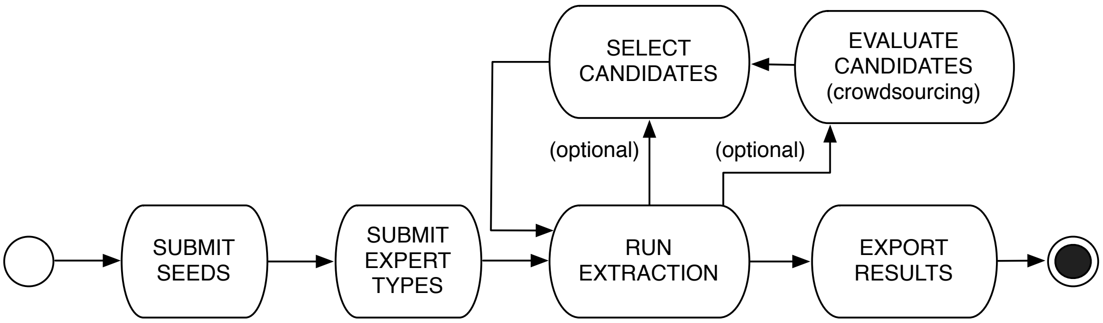


Figure 3.1: *Iterative knowledge extraction process.*

(a) *Export* them (in CSV format for human consumption or data analysis purpose or in RDF format for further integration in existing semantic knowledge bases);

(b) Forward them to *domain experts or a generic crowd for result evaluation* purposes (validation);

(c) Use them (or a subset of them) as new seeds and *iterate the whole pipeline*.

3.2.3 Example Domains of Interest

We applied our method on eight heterogeneous domains and usage scenarios (two of them already described in the previous section), so as to demonstrate its generality:

- **Fashion designers**: the research team of the Fashion In Process Lab[4] (especially Paola Bertola, Chiara Colombi and Federica Vacca) was among the inspirators of the previous work, as it brought to us the problem of identifying emerging fashion designers. In the original experiment, the domain experts started with 200 emerging Italian brands as seeds.

- **Finance influencers**: a team of economics and statistics researchers at University of Pavia executed experiments on the extraction of influencers in finance. In this case the team selected as seeds 120 bloggers and journalists in the finance sector.

- **Fiction writers**: We considered some fiction authors engaged in the Melbourne Emerging Writers Festival[5] by picking 20 seeds from the participants to the event.

- **Craft breweries**: We considered as seeds a set of 20 well-known US craft breweries, all present in DBpedia.

- **Chess players**: We used a list of 20 top chess players and their accounts.[6]

[4] http://www.fashioninprocess.com/

[5] http://www.emergingwritersfestival.org.au

[6] https://www.reddit.com/r/chess/comments/32t5ov/list_of_top_chess_player_journalist_twitter/

- **Jazz players**: We used a list of 10 top jazz players and their accounts.[7]
- **Fashion models**: We used a list of 20 fashion top models.
- **Talk shows**: We used a list of 20 official Twitter accounts of popular TV talk shows.

These scenarios cover different information needs and domains. For instance, fashion design is characterized by a very high concentration of the domain in few brands only, most of which well known; on the opposite, fiction writers is an open domain where authors can be considered widespread; finance is a well established domain with renowned influencers; and craft beer is experiencing a tremendous growth with new craft breweries emerging almost daily.

3.2.4 Extraction Scenarios

At the purpose of responding to the four research question presented in Section 3.2.1, we describe four possible usage scenarios for our method, and we report the findings obtained by experimenting with the scenarios on the eight domains discussed above.

Iterative Knowledge Extraction

The first usage scenario we want propose is *iterative* knowledge extraction, where successful candidates of one extraction are used as seeds for a subsequent extraction. We identified 20 seeds per domain, and ran 3 iterations of the method for each of them.

Table 3.2 shows the precision@10 and precision@20 obtained for each extraction for the eight domains; the #seeds in the second and third run correspond to the candidates of the respectively first and second run that were considered correct among the top 20 candidates (#results is the number of all candidates identified in a given run). Correctness was assessed against a manually tagged ground truth built through crowdsourcing; each run was executed twice. Every run after the first takes the good candidates of the previous run as seeds.

Table 3.2: *Precision @10 and @20 of iterative knowledge extraction experiments using candidates produced in one run as seeds of a consecutive run; #results are the overall identified candidates.*

	RUN #1				RUN #2				RUN #3			
SCENARIO	#seeds	#results	Pre@10	Pre@20	#seeds	#results	Pre@10	Pre@20	#seeds	#results	Pre@10	Pre@20
Fashion Designers	20	407	0.3	0.3	6	282	0.1	0.25	5	295	0.2	0.15
Fiction Writers	20	426	0.7	0.55	11	435	0.4	0.5	10	439	0.3	0.55
Chess Players	20	418	0.7	0.5	10	389	0.7	0.6	12	424	0.6	0.6
Finance	20	777	0.5	0.3	6	432	0.3	0.45	9	514	0.4	0.45
Craft Breweries	20	972	0.1	0.25	5	240	0.1	0.1	2	128	0.4	0.3
Jazz Players	20	428	0.8	0.8	15	431	0.8	0.8	16	426	0.9	0.85
Fashion Models	20	413	0.1	0.2	4	138	0.1	0.2	4	211	0.4	0.35
Talk Shows	20	423	0.5	0.45	9	440	0.3	0.45	9	437	0.4	0.35

[7] http://oneworkingmusician.com/10-jazz-musicians-you-should-follow-on-twitter/

Within a given domain, consecutive runs tend to produce similar precision, independently of the number of seeds and results. It seems that certain domains are most suited to the method, such as chess or jazz players, most likely because the twitter accounts of these entities are focused on (if not limited to) their respective domains, whereas the method is less effective in other domains, such as breweries or fashion models. This latter result may be due to tweets that are less focused and contain generic topics, making similarity search less effective, but also to the presence of entities with high similarity but different ontological types (e.g., beer lovers/distributors or fashion bloggers). If initial entities are chosen from a specific subdomain (e.g., writers in Melbourne), iterations progressively extract entities from a wider semantic and geographic domain (e.g., from outside Australia).

If we consider all runs as independent (considering neither the domain nor the order of execution), we find a correlation of 0.65 between the number of seeds and that of results (at the edge of significance) and one of 0.91 between precision at 10 and precision at 20. If we analyze the domains individually, pair-wise t-tests among the three runs neither identify any significant difference ($\alpha = 0.05$) between precision at 10 nor between precision at 20. The method thus works well even after several iterations, as precision remains rather stable (it decreases in certain domains, but it also increases in others); hence *a recursive application of knowledge extraction methods finds an increasing number of domain entities* **(RQ1)**. Especially when precision is high, one can find a good number of correct emerging entities from within the list of top-20 candidates.

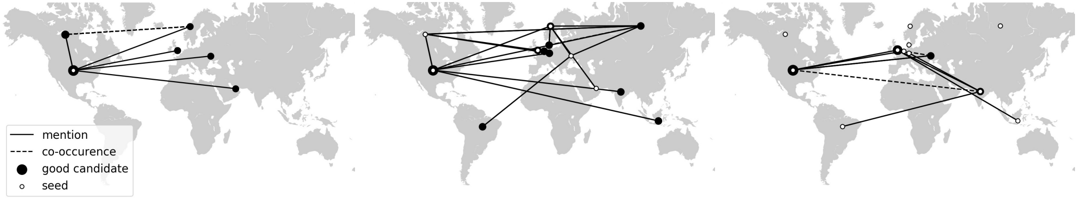
Geographical Spreading of Knowledge

In order to study the geographical spreading of knowledge, we applied a similar iterative knowledge extraction approach as in the previous section to one selected domain: chess players from the US. We decided to focus on this sub-domain (of all chess players) to study if our method can find entities from other geographical regions and, if yes, how fast the knowledge graph expands.

The experiment lasted three runs. For the first run of the experiment we took 7 seeds and a set of expert types. The next two re-runs were performed selecting the correct candidates from the top 20 results of the respective previous run. The actual localization of candidates was performed manually, either using the declared Twitter user location or, if that was missing, by searching other social resources and matching entities. After careful study of the location field as used by different Twitter accounts, we set the granularity of the locations to the level of individual countries.

The result is an instance-based graph of *mentions* from a seed to a candidate and *co-occurences* of two candidates, i.e., tweets by one of the seeds that mentioned the two candidates together, mapped to physical locations. The result is illustrated in Figure 3.2.

The first knowledge extraction produced 12 good candidates from different countries and continents, reaching Europe and the Middle-East. The first iteration found 15 good candidates, adding new data points also to South America and Asia. The second iteration produced 10 good candidates, even if some seeds did not find any valid candidate.



(a) First run (b) Second re-run (c) Third re-run

Figure 3.2: *Geographical dispersion of the knowledge graph in response to iterative knowledge extractions (US chess players).*

We conclude that *discovered knowledge which is iteratively found spans large geographical areas very fast* **(RQ2)**. The finding is somewhat surprising, but can likely be explained with the open nature of Twitter, e.g., compared to Facebook (where we would expect a slower spreading).

With this experiment we proved that even starting from a small set of seeds from a single country, our approach is able to find good candidates from different states, countries and continents.

Capturing of Knowledge Evolution

The third usage scenario we propose is the study of how knowledge evolves over time. While the previous two scenarios are instances of an iterative knowledge extraction process, with selected candidates being used as seeds, here we propose a *periodic* extraction process, with knowledge extracted at periodic time intervals. For convenience, we fix the interval to three months, starting from September 2017 and looking back until January 2016. At each period, we consider all tweets since the beginning of 2016 up to the last month of the period being studied, constructing smaller reference data sets as we go back in time. It is important to note that to go back in time all cut-offs are computed from one cumulative download of tweets performed in the end of September 2017 using one set of seeds.

Table 3.3 reports the numbers of candidates extracted for four domains. The four domains have a different evolution over time, with *Finance* being the youngest domain (our seeds started tweeting only in 2017). For the other three domains, one can observe that the *Fashion* domain growing slower than both *Chess* and *Australian Writers*. Looking at the table, it is also important to note that the rate at which knowledge increases is fast, that is, the knowledge we extract today is significantly bigger then the one we would have extracted only 3 months ago. Projected into the future, this solicits

Table 3.3: *Looking back in time in the four domains: periodic knowledge extractions over a period of 21 months.*

Time interval	Chess	Finance	Writers	Fashion
2016/01 - 2017/09	545	151	780	153
2016/01 - 2017/06	310	52	329	123
2016/01 - 2017/03	210	45	237	103
2016/01 - 2016/12	146	0	177	95
2016/01 - 2016/09	78	0	94	79
2016/01 - 2016/06	43	0	45	61
2016/01 - 2016/03	10	0	25	27

Table 3.4: *Emerging knowledge compared to Wikipedia among the correctly identified candidates.*

Domain	Emerging entities
Fashion Designers	100%
Finance	77%
Chess Player	42%
Australian Writer	36%

a continuous knowledge extraction instead of a periodic or random extraction. In conclusion, *knowledge can be extracted from social data at arbitrary points of time in the past and it is possible to trace how knowledge will evolve in the future*, thanks to the possibility to extract knowledge continuously **(RQ3)**.

Identification of Emerging Knowledge

For the analysis of how much knowledge reconstructed from social content can be considered as *emerging* (low-frequency entities not yet included in generic ontologies with high-frequency knowledge), we refer to *Wikipedia* as generic source of knowledge. We performed the analysis over four domains (*Fashion designers*, *Finance*, *Chess Players* and *Australian Writers*) we took the candidates produced with *one* iteration of the method and calculated the percentage of correctly identified candidates. To assess this aspect, we proceeded by counting how many candidates have a Wikipedia page, i.e., had already been captured formally.

Table 3.4 plots the obtained results. These are very domain dependent, likely due to the different social context behind the domains. For instance, in the *Fashion Designers* domain, the method produced an unexpected 100% of emerging designers. Instead, the domain that produced the lowest number of emerging entities is Australian writers (36%). Despite these fluctuations across domains and the fact that the reported results may not grant statistical representativeness, it is however important to note that in all cases *knowledge extracted from social content using the described method includes some relevant emerging knowledge that can be added to ontologies* **(RQ4)**.

3.2.5 Implementation

The proposed approach has been implemented as a Python application. With respect to the original research [47], which extensively investigated more than 900 alternative extraction strategies, in this work we propose a light-weight tool, which only applies one strategy (the best one in [47]). While the quality of the results slightly decreases, the tool performance is good, in terms of tweets download, DBpedia matching, and score computation; the performance is adequate for exploring many domains on daily basis. The tool can run continuously or with periodic iterations, using the results as new seeds.

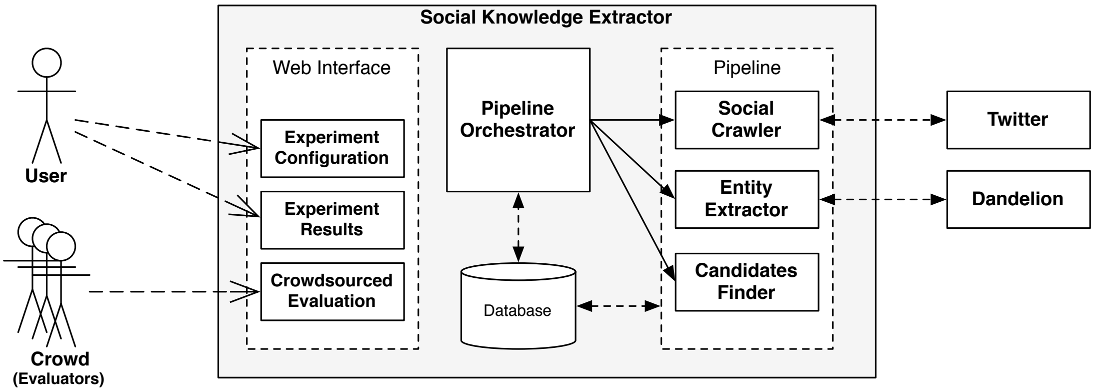


Figure 3.3: *Architecture of the tool implementing SKE.*

Figure 3.3 represents the high-level view of the **system architecture**. The *Web Interface* allows users to interact with the system: it supports the phases of experiment definition and results visualization by the expert, as well as the validation of the results by the crowd. The *Pipeline Orchestrator* manages the execution of the process and is responsible of coordinating the components that perform each step of the analysis. The involved components are the following:

- *Social Crawler*: this component receives in input a list of Twitter handles (i.e., user identifiers) and uses the Twitter API [8] to crawl their tweets. It is used for retrieving the posts of both the seeds and the candidates;

- *Entity Extractor*: this component receive in input the text of the tweet and uses the Dandelion API[9] to find entities mentioned in the text. Dandelion is a commercial software which matches a text to either instances or types of DBpedia;

- *Candidate Finder*: this component is responsible of ranking the candidates using the information retrieved by the other components. In particular it creates the feature vectors and computes the similarity score of each candidate.

[8] https://dev.twitter.com/rest/public

[9] https://dandelion.eu/

The data involved in the process is persisted in a MongoDB[10] database, which stores, for every user, the track of all his experiments, in terms of seeds, candidates, and evaluations.
The code is available online on GitHub under the Apache 2.0 open source license.[11]

3.2.6 Related Work

This section presents a method and tool to harvest the collective intelligence of the Social Web in developing a collective knowledge system [138].

P. Mika pioneered this area in [202], by identifying broader and narrower terms using social network analysis methods such as centrality and other measures like the clustering coefficient. Our interest is on the circle of knowledge life proposed in [268], where emerging knowledge is extracted from the Social Web using known facts captured in a knowledge graph.

Our approach is grounded in homophily, a key aspect of social networks: entities are related when they have similar characteristics , as shown by Barabási and Albert [16]. Homophily can be used to explain the scale-free nature of social networks; in our approach, the seeds guide the process that identifies homophily patterns and thus constructs the domain graph.

As pointed out by Weikum and Theobald [307] and by Etzioni et al. [110], the grand challenge in automating the discovery of emerging knowledge is to find entities, relationships and attributes not mainstream, belonging to niches in the long tail (A. Chris [65]).

We found two works that also proposed to use Twitter for ontology enrichment. P. Monachesi and T. Markus in [210] proposed an ontology enrichment pipeline that can automatically enrich a domain ontology using data extracted by a crawler from social media applications.

C. Wagner and M. Strohmaier [302] investigated a network-theoretic model called tweetonomy to study emerging semantics. Complementary to our work, they investigated how the selection of tweets (so-called Social Awareness Streams) can lead to different results.

3.2.7 Conclusion

In this section, we explored a method for discovering knowledge from social media. The approach is iterative, and entities produced by one iteration are the seeds for the next iteration; we considered eight domains. We specifically described the geographic and temporal spreading of entities extracted by the method. Finally, we measured the number of emerging entities found; we define the entities not present in Wikipedia as

[10] https://www.mongodb.com
[11] https://github.com/DataSciencePolimi/social-knowledge-extractor-2

emerging knowledge. I show that this method achieves a high precision after several iterations in many domains, particularly when analyzing chess and jazz players. In such domains, the terms used in social communications are the most domain-specific.

Since the main component of the knowledge extraction pipeline is the computation of similarities between users, in the next chapter, I investigate this stage in detail. My main goal is to find the best approach to encode users into feature vectors that reflect our idea of similarity. I investigate communities of users, controversies between communities and how Language Models help to compute semantically similar tweets and users.

CHAPTER 4

Community characterization, Controversy Detection and User Similarity

A community is a set of people with something in common. The features that link people are not defined *a priori*, but different selections lead to diverse clustering of users in communities. For example, if we define communities as people connected by blood ties, the population will be divided into families. However, if we define communities based on shared interests, we will obtain a completely different partition, where we will group people playing tennis, people singing, people painting and so on.

Usually, community detection algorithms have been developed to cluster nodes in graphs based on their links. Given a graph of people connected by friendship bonds, we can group people obtaining clusters of friends that share something, e.g. same class in college or same sports team. Recent researches propose many algorithms to detect clusters of nodes in graphs, focusing on the best trade-off between computational speed and accuracy.

However, in this chapter, I do not think about communities as clusters of nodes in social networks since I do not inspect the underlying graph of users, but I define a community of people as the set of people with similar interests, even if not *explicitly* connected. This definition leads to a broader sense of community, where users that do not know each other can still be part of the same community, e.g. chess players could have never met, but they still belong to the same international community.

In the first work of this chapter, I investigate how to characterize communities based on the vocabulary they use when posting on Twitter. I define a method to calculate the strength of a community and a membership criterion, to check whether a user belongs to the community. To find the best approach, I test numerous combinations of syntactic and semantic features of users belonging to three different communities: fashion designers, Australian writers and chess players.

In the second section of this chapter, I design a content-based approach to detect controversies in social networks. I aim to automatically classify if a topic is controversial or not by looking at what users post. I apply Language Models to embed tweets, and I compare them to understand if the users belong to two or more groups fighting each other or if they have similar opinions about the selected topic.

The third work describes how to exploit Twitter as a source of large corpora of textual documents used as training sets for semantic sentence embeddings. The trained Transformer-based models reach state-of-the-art results in semantic textual similarity tasks when data from social networks are involved. I perform the training with a triplet-like loss. However, the selection of input documents is limited in length by the nature of the model, allowing analysis at the tweet level.

The last work solves the length issue with the introduction of a Hierarchical approach. Users are embedded into feature vectors by Transformer-based models designed to process single tweets (the same model described in the previous work) and then merge the tweets embeddings into a single dense high-dimensional vector. The design of the training and evaluation datasets make the whole approach statistically significant and completely reproducible. Finally, I inspect the embeddings to check whether they reflect our idea of similarity. We perform community visualization, outlier detection and controversy detection.

4.1 Content-based Characterization of Online Social Communities[1]

4.1.1 Introduction

Defining the essence of a community is difficult: in the English dictionary, a community is the *condition of having certain attitudes and interest in common*. The concept of community is general and goes beyond social networks and Internet, but finding communities in the digital world is very relevant, as it has a huge number of social implications and potential commercial exploitations [162, 190, 225].

Digital social content can be automatically inspected, hence, social communities on Internet can be detected by algorithms [223, 225, 260]; this process comes with very interesting challenges from a social analysis perspective, as well as interesting computational problems. Social networks can be considered as big graphs of linked nodes; most methods for community detection use as initial input the arcs among actors [117] (e.g. the *friendship/follow* relationships), or take into account social activities [260] (e.g., the *likes* or *comments*). These methods build weighted graphs representing social interactions and then look for subgraphs with certain properties (e.g., the sparsity/density of subgraphs), typically corresponding to subsets of highly interacting users.

In this Section, we explore a different direction, and propose a **content-based approach to community detection**. We conjecture that a community can be characterized by the content that they share, as it is a very strong distinctive property. With this approach, we define simple methods for community detection: given a set of social actors, we argue that they form a community if their shared content has strong similarity properties; we can also test if a social actor is a member of a community by comparing the actor's content to the community's content. As we will see, content-based analysis can be performed bottom-up, with very few actors forming an initial community, and thus it is less computationally demanding than link-based analysis.

This work is part of a general effort towards the use of social accounts for extracting semantic knowledge; in particular, in [47] (section 3.1) we defined a method for extracting emerging knowledge from social accounts based on co-occurrence of accounts with known members of a community; in [45] (section 3.2) we observed that very few accounts are sufficient to generate a community and we explored how such community grows in space and time as effect of iterative applications of the method. In this work, we concentrate on a systematic study of social content features that best characterized a community. Preliminary work [246] considered fewer textual features (in particular, no latent feature) and fewer contexts of application; this Section summarizes also that work including also new latent features that actually have the best performance in the

[1] Authors: Giorgia Ramponi, first author, conceptualization, implementation, experiment design, writing; Marco Brambilla, conceptualization, editing; Stefano Ceri, conceptualization, writing, editing; Florian Daniel, conceptualization, editing; Marco Di Giovanni, first author, conceptualization, implementation, experiment design, editing. [247]

new contexts.

To better define our approach, we consider Twitter as social network and we study the communities of Twitter accounts; with this method, every Twitter account is associated with several tweets, and we consider the vocabulary of terms used in their tweets[1].

We define the following problems:

- Given a community of n Twitter accounts, define the *strength* of the community, measuring how the community is well characterized by the shared vocabulary of its members.
- Given other accounts, define *membership criteria* for deciding if they are also part of the community.

Solving these problems requires addressing two challenges.

- Selection of textual features: as Twitter typically uses short sentences and has its own given jargon, we must choose among syntactic or semantic elements of the *Twitter jargon*;
- Measuring the distance between features associated to accounts, so that we can test community's strength and membership.

The research question underlying these challenges is to ascertain how much communities can be guessed by considering just the content of their social interaction. We will consider a variety of options for both challenges, but we will eventually see that simple choices work remarkably well in practical contexts, suggesting that this approach has a wide applicability.

Although our approach applies to possibly large communities (e.g., the followers of politicians, as shown in Table 4.9), it is best suited to the characterization of small communities with highly specialized vocabulary, where the method performs remarkably well; problems that exhibit these features have significant applicability, discussed later.

This Section is organized as follows. In Section 4.1.2 we define the metrics used later: dispersion and coherence. In Section 4.1.3 we define the syntactic and semantic features used to perform the analysis and the methods for extracting them, while in Section 4.1.4 we select the most effective features for testing a community's strength and membership. In Section 4.1.5, we assess the power of content in two important applications related to detection of communities in the political arena and to targeted advertising. We present related work in Section 4.1.6 and conclusions in Section 4.1.7.

[1] The approach can be easily transferred, e.g. on Facebook (and other social networks) by using accounts and posts.

4.1.2 Background

Definitions

We introduce some useful definitions in the community detection problem:

Community: a community C is a set of Twitter accounts that have one or more characteristics in common;

Member: a Twitter account of a community;

Candidate: a Twitter account that could be included in the community;

Feature Vector: we associate to every member or candidate c a *feature vector* $f_c =< f_{c,1}, f_{c,2}, .., f_{c,n} >$, whose elements are the frequencies of the textual features (TF in Section 2.2.2) extracted from a corpus consisting of the last 200 tweets of c. Thus, if for example we are considering *nouns*, $f_{c,i}$ is the frequency of use of the noun i in c's tweets.

Centroid: given m feature vectors $\{f_1, ...f_m\}$ of cardinality n, we define the centroid:

$$z =< z_1, .., z_n >$$

where:

$$z_i = \frac{1}{m} \sum_{c=1}^{m} f_{c,i}$$

Distance metrics

To evaluate the closeness of a candidate c to the centroid z we consider four distance metrics:

- Manhattan distance ($L1$): $d_{L1}(c, z) = \sum_{i=1}^{n} |c_i - z_i|$
- Euclidean distance ($L2$): $d_{L2}(c, z) = \sqrt{\sum_{i=1}^{n} (c_i - z_i)^2}$
- Cosine distance: $d_{cos}(c, z) = 1 - \frac{\sum_{i=1}^{n} c_i z_i}{\sqrt{\sum_{i=1}^{n} c_i^2}\sqrt{\sum_{i=1}^{n} z_i^2}}$
- Kullback-Leibler Divergence (KLD), also called relative entropy, is a metric of how one probability distribution diverges from another:

$$d_{KL}(c, z) = D_{KL}(c||z) = \sum_{i=1}^{n} c_i \log(\frac{c_i}{z_i})$$

Dispersion Index

The dispersion index is a measure of the cohesion of a community. We consider the ratio D_c/D_T, where D_c is the average distance of the members of the community to the community centroid; D_T is the average distance of the members of the community to the centroid of the vocabulary used by all Twitter accounts. We expect a dispersion index between 0 and 1, where a smaller dispersion index is associated to communities with stronger cohesion.

Coherence metric

We can define the coherence of a text as a “continuity of senses" [255] which requires arguments to be logically connected. In topic modeling, a coherent model [38, 216] is capable of describing a set of topics in a rigorous way. Measuring coherence is a complex task, but we refer to the work of [255] which provides a systematic study on different coherence measures, and proposes C_V as the best one.

C_V is obtained by evaluating all the possible combinations of four different dimensions and picking the one that performed best on a given dataset evaluated by humans:

1. type of segmentation used to divide the word set into subsets: C_V uses a *one-one* approach, where every pair of words is selected;

2. how probabilities are derived: C_V uses *Boolean Sliding Window* with window size of 10. The probability is calculated as the number of windows in which the word occurs divided by the total number of windows;

3. Confirmation Measure, defining a way to compute how strong a word set supports another one: C_V uses *indirect cosine measure* to calculate cosine similarities between vectors obtained with the direct *normalized log-ratio* measure;

4. aggregation of all subset scores to a single score: C_V uses the *simple average* of all the values.

4.1.3 Content Features Description and Creation

Syntactic Features

Words in tweets are classified on the basis of their syntactic features and recognizing, in particular, verbs and nouns. Syntactic analysis consists in associating them with their frequency in the tweet corpus.

The process starts with a standard text pre-processing of texts by removing stopwords, tokenizing and tagging the text and retrieving the root form of the words, using the NLTK library [28]. After pre-processing, we select words carrying only three different tags: nouns, verbs and proper nouns (a subset of nouns). The syntactic features are Term Frequencies (TF in Section 2.2.2) of the selected words.

Semantic Features

The meaning of each word in a language is formed of a set of abstract characteristics known as semantic features. Every language is associated with a hierarchical structure representing semantic features, typically words are at the leafs of these hierarchies and semantics is assigned by traversing the hierarchy. When we consider semantic features, we go beyond the word itself, by extracting its meaning.

In our work we used two kinds of semantic features: knowledge-based features, and topic features, obtained by using topic detection techniques.

Knowledge-based features are extracted after text matching with a structured knowledge graph; since we do not set a specific domain of interest, we select DBpedia[2], which is publicly available and easily accessible through APIs; it provides structured content from the information created in Wikipedia[3].

In order to extract semantic features from tweets we pick Dandelion[4], a commercial software which matches a text to DBpedia entities. We consider a term as semantically understood when it is matched to either a type or an instance, defined as follows:

- *type*: a *type* is an element of the DBpedia hierarchy; Dandelion produces matches with associated probability and we use the default threshold value (0.6)[5];
- *instance*: some words are also associated to a concept that has a page in Wikipedia; we call these concepts *instances*.

The semantic features are Term Frequencies (TF) of types and instances.

Topic features are learned using the Latent Dirichlet Allocation (LDA) process [30]; the process learns the relations between words in documents and creates a fixed number of topics; each topic, in turn, is associated with a probability distribution Φ over the words that are recognized as significant for that topic.

To consolidate the use of LDA in our context, we have to decide how to set an ideal number of topics, which is a prerequisite of the method. We consider the corpus of tweets of a specific domain and divide it into a training and testing set. We build 50 different models, each one with an incremental number of topics (from 1 to 50), and for each of them we calculate the C_V coherence (Section 4.1.2). Then we select the number of topics yielding to a model with the highest value of coherence. In most corpuses of Tweets, the best coherence value found is small, in the selected domains, it ranges between 4 and 10.

[2] https://wiki.dbpedia.org
[3] https://www.wikipedia.org/
[4] *https://dandelion.eu*
[5] The threshold is for the confidence value of the annotation extraction

Given a specific tweet, LDA associates it with a probability distribution over the topics. We use this probability distribution as topic features vector. We implement LDA with the Gensim library [251].

4.1.4 Evaluation

We formulate the problem of *finding the best set of features and the most effective distance metric in order to characterize community membership.* Given a community $C^* = \{c_1, ..., c_n\}$, we retrieve the tweets of these accounts and construct one feature vector for each of the six textual features discussed above. From these feature vectors, six centroids z_{type}, $z_{instance}$, z_{noun}, z_{verb}, $z_{propernoun}$ and z_{topic} are created.

We explore which combination of textual features and distance metrics achieves the best result in predicting that a candidate account c_i is a member of the community and that the community is strongly or weekly characterized.

The experiment is artificially built by starting from known community members and separating them into two sets, one of which is merged with randomly selected accounts. We use the alternative features and distances, measure their effectiveness in ranking the top candidates, and select the features and distances associated with the best rankings.

Input Data and Experiment Design

We consider three initial communities of twenty well-characterized professionals, each member of a specific domain as defined by domain experts, that constitute our gold standard. Accuracies are highly dependent on the domain, thus there are communities harder to characterize because their vocabulary is less specialized.

The communities are formed by fashion designers, Australian writers, and chess players:

- **Fashion designers**: the research team of the Fashion In Process Lab[6], in the original experiment, collected emerging Italian brands, and we used 19 of them;
- **Australian writers**: we considered some fiction authors engaged in the Melbourne Emerging Writers Festival[7] by picking 20 accounts from the participants to the event;
- **Chess players**: we used a list of 20 top chess players and their accounts[8].

For every Twitter account we select at most the last 200 tweets, which correspond to a single Twitter API call; exact sizes are reported in Table 4.1. Data have been collected on 08/02/2018. The anonymized dataset is available and can be downloaded at `https://doi.org/10.7910/DVN/VWLEAA` [46].

[6] http://www.fashioninprocess.com

[7] http://www.emergingwritersfestival.org.au

[8] https://www.reddit.com/r/chess/comments/32t5ov/list_of_top_chess_player_journalist_twitter

Table 4.1: *Sizes of Datasets*

	Number of users	**Number of Tweets**
Fashion designers	19	1536
Australian Writers	20	1953
Chess Players	20	2262

Experiment Design

For every community, we consider ten Twitter accounts as community members; we consider a set of candidates constituted by the other ten members and by 160 random accounts. We repeated each extraction 50 times, and averaged the performance indexes.

For every choice of domain, features and distance, we compute the centroid of the ten community members and we rank all the candidates in terms of distance from the centroid. We also compute the number of topics yielding the best coherence. We consider *precision@10* and *recall@20* as relevant performance indicators; the experiment goal is to retrieve the known ten members of the community within the top-ranked candidates.

Comparison

Table 4.2 shows the results of our experiments. By comparing the four distances, we notice that KLD and cosine distance provide the best results in terms of precision and recall in every domain, therefore we focus on them. Overall, the best syntactic feature is *proper noun (NNP)* while the best semantic ones are *Instance* and *Topic*. *Instance* obtain comparable results to *Topic* and *NNP*, but its extraction requires an interaction with the commercial software Dandelion, whose free use is limited in rate, so we exclude it from our further analysis.

By comparing the domains, we note that precision and recall are generally higher for Chess Players, intermediate for Fashion Designers, and lower for Australian Writers. In particular, precision is extremely good for Chess Players, where all methods find the first 6 members as top ranked among all 170 candidates; and it is rather good for all domains, including Australian writers, as we find 4 members within the top ten ranked.

Dispersion Indexes

We inspected the Twitter accounts of chess players, and we found that chess players tweet almost exclusively about chess, hence their vocabulary is narrower and most focused; fashion designers talk a lot about fashion but they also talk about several other close topics; and Australian writers intertwine tweets about writing with tweets about many other topics, including personal experiences. This empirical consideration is quantified by using the dispersion index measuring the internal coherence of a community, defined in Section 4.1.2, whose values for the three communities are summarized

Table 4.2: *Exhaustive analysis showing the precision@10 and recall@20 for experiments built by combining in all possible ways four choices of distances and seven choices of features in three domains. We use labels CD for cosine distance, KLD for Kullback-Leibler Divergence, l1 for Manhattan distance and l2 for Euclidean distance.*

Domain	**Feature**	**cd**$_{precision}$	**cd**$_{recall}$	**KLD**$_{precision}$	**KLD**$_{recall}$	**l1**$_{precision}$	**l1**$_{recall}$	**l2**$_{precision}$	**l2**$_{recall}$
Chess	NNP	0.800	**0.905**	0.770	0.870	0.800	0.885	0.140	0.270
	Noun	0.270	0.335	0.690	0.825	0.660	0.795	0.165	0.215
	Verb	0.155	0.235	0.130	0.330	0.200	0.350	0.135	0.200
	Instance	0.835	0.875	0.775	0.860	0.750	0.810	0.320	0.385
	Type	0.385	0.430	0.700	0.785	0.420	0.560	0.360	0.410
	Topic	0.726	0.824	0.702	0.834	0.734	0.868	0.732	0.822
Fashion	NNP	0.510	0.695	0.560	0.745	0.625	0.690	0.001	0.040
	Noun	0.180	0.345	0.485	0.610	0.710	0.770	0.075	0.150
	Verb	0.010	0.030	0.100	0.105	0.070	0.105	0.010	0.015
	Instance	0.695	0.765	0.595	0.765	0.705	0.750	0.001	0.015
	Type	0.120	0.250	0.165	0.195	0.235	0.315	0.125	0.240
	Topic	0.780	**0.870**	0.736	0.816	0.654	0.764	0.656	0.748
AW	NNP	0.245	0.435	0.265	0.385	0.310	0.450	0.030	0.030
	Noun	0.095	0.130	0.075	0.220	0.200	0.415	0.110	0.170
	Verb	0.120	0.190	0.005	0.155	0.085	0.190	0.115	0.165
	Instance	0.390	0.515	0.335	0.560	0.245	0.415	0.075	0.115
	Type	0.110	0.245	0.095	0.190	0.165	0.250	0.110	0.230
	Topic	0.522	**0.642**	0.444	0.570	0.406	0.532	0.378	0.484

in Table 4.3 (a high index is indicative of high dispersion).

Table 4.3: *Dispersion index for the three domains.*

Features	**Domain**		
	AW	Fashion	Chess
NNP	0.84	0.79	0.55
instances	0.80	0.73	0.63

Topic Explanation

Topics are explained by their most recurrent words; in Table 4.4 we report the first 5 words explaining the first topic for each of the three domains. As we can see, in Chess players the best topic contains the word *chess* and *game*; the best topic for Fashion contains the word *love*.

Table 4.4: *Best topics that represents the three domains with their first 5 components.*

Domain	**Topics**				
Chessplayers	raider	italy	owner	chess	playoff
Fashion	day	time	thank	get	love
AW	person	thank	time	thing	way

Conclusion of the Evaluation

After this analysis, we conclude that the best features are proper nouns and topics (associated with any distance). The former is a syntactic feature, describing terms which denote concrete aspects of reality; the latter is a latent semantic feature, representing the texts in their entirety.

4.1.5 Applications

In this Section we propose two applications, showing that each selection can be the most useful for characterizing specific social communities.

Content-based Analysis of Accounts from a Political Perspective

One of the most interesting applications of content-based community detection is concerned with understanding political preferences. Politics is most influenced by the use of social media, as many politicians deliver their comments using Twitter. We therefore asked ourselves if the use of vocabulary could be suggestive of political preferences. At the March 2018 elections in Italy, three coalitions participated to the competition: the Right parties, Cinque Stelle, and the Democratic Party. We considered some politicians from the three coalitions, and we retrieved their last tweets (a single Twitter API call per user). We performed the following experiments:

- We used as before a limited number of accounts as community members and we classified the remaining accounts on the basis of their similarity to the centroid; we repeated this experiment 50 times, every time selecting randomly the accounts to use as community members. Data have been collected on 18/04/2018.
- We repeated the test by using the followers. In this case, as we assume that the follower of a politician prefers the politician's party, we developed a predictor of the political preferences of the followers based on the vocabulary used. We considered the followers of politicians of just one of the three coalitions, thereby excluding those followers who observe politics from a neutral perspective (e.g. journalists). Data have been collected on 06/05/2018.

Sizes of the datasets are reported in Table 4.5. Results of the first experiment are presented in Table 4.6. The method is extremely accurate in classifying the accounts of the elected politicians, suggesting that indeed they have a very different vocabulary.

Table 4.5: *Sizes of Political Parties Datasets*

	Number of users	**Number of Tweets**
Right parties	19	2174
Cinque Stelle	20	2295
Democratic Party	25	3452
Right parties followers	126	4948
Cinque Stelle followers	289	16145
Democratic Party followers	306	17201

Table 4.6: *Prediction of parties of members of the Italian parliament using proper nouns.*

	Right Parties	Cinque Stelle	Democratic Party
Right Parties	99.68%	0.0%	0.32%
Cinque Stelle	0.00%	100.00%	0.00%
Democratic Party	0.00%	0.00%	100.00%

Table 4.7: *Prediction of parties of the followers of politicians using proper nouns.*

	Right	Cinque Stelle	Democr.
Right parties followers	96%	0	4%
Cinque Stelle followers	0	40%	60%
Democratic Party followers	0	0	100%

In Table A.2 in Appendix A we report the most frequent proper nouns for the three parties. As you can see it is not easy to interpret this feature because proper nouns are too specifically connected with factual people, location or events occurring in Italy. Consider for instance that top mentioned proper nouns include Bologna, Milano, Calabria for Democratic Party, Friuli for the Right Party, Roma and Torino for Cinque Stelle, and these are locations where each party is either historically strong or actually at the local government.

To show the different vocabularies between parties we present in Table A.3 in Appendix A most frequent nouns, that are slightly less effective than proper nouns in characterizing communities, but can be best perceived by readers based upon general knowledge. The three lists have many common terms in any conversation (e.g. day, year) or in any conversation of politicians (e.g. “government, job, program, country", or “law, citizen" appearing in two lists out of three) and at first sight look very similar; but if one looks at terms which appear just in one list, finds “Italian, tax, security" in the Right Party, “movement, live" in Cinque Stelle and “campaign, woman, club, commitment" in the Democratic Party; we can clearly see that the different vocabulary characterize the parties.

Results of the second experiment, reported in Table 4.7, are rather surprising and have an interesting sociological interpretation. We note that the method correctly predicts the followers of the Democratic Party (100% accuracy) and of Right Parties (96% accuracy). For what concerns Cinque Stelle, however, the predictor only achieved 40% accuracy, while it classified the followers as politically closer to the Democratic Party

ht

Table 4.8: *Prediction of the parties of members of the Italian parliament using topic features.*

	Right Parties	Cinque Stelle	Democratic Party
Right Parties	52%	17%	31%
Cinque Stelle	53%	24%	23%
Democratic Party	48%	26%	26%

Table 4.9: *Prediction of the followers of politicians of the three parties.*

	Right	Cinque Stelle	Democr.
Right parties followers	52%	17%	31%
Cinque Stelle followers	16%	17%	66%
Democratic Party followers	14%	16%	70%

(60%) and not to the Right Parties (0%). This is an indication that the followers of Cinque Stelle do not have a distinctive vocabulary, and have stronger similarity to the Democratic Party than to the Right Parties. These results are confirmed by the dispersion indexes, which show stronger dispersion for Cinque Stelle (see Table 4.10).

Table 4.10: *Dispersion index for the followers of politicians of the three parties.*

	Right	Cinque Stelle	Democr.
dispersion index	0.34	0.58	0.48

We repeat the experiment using topics as features. As we can see in Table 4.8 for the first analysis and in Table 4.9 for the second analysis, the results are not satisfying, as the method does not succeed in classifying political parties. A likely reason is that, while nouns are very indicative of a party, topics are not, as tweets written by politicians end up having the same topics regardless of their party.

Targeted Advertising

From a commercial point of view, the most important application of community detection is targeted advertising. We assume that the advertiser already knows a community of interest, e.g. thanks to activities that the community has already performed in controlled social platforms. The advertiser's objective is to enlarge the community by finding new candidate accounts, thus potential new customers.

Among the many possible examples of applications, we consider sport events, in particular baseball or football events, where we initially know a set of accounts of players of those two sports. In such case, the advertiser's interest is to broaden the set of accounts that she can reach by adding similar accounts to the initial set. Following a pipeline similar to the one described before, we manually collected Baseball players and Football players of UCF (University of Central Florida), and randomly split them in a set of 10 accounts that represents the already known community, and a set of accounts that we expect to retrieve when mixed with random Twitter accounts. In Table 4.11, the sizes of the datasets are reported. Data have been collected on 22/02/2018. The anonymized dataset is available at `https://doi.org/10.7910/DVN/VWLEAA`. In Table 4.12 we compare the results obtained when using NNP and topic features, using the cosine distance.

In this case, topic features achieve the best performances in the two communities, as the community of baseball players and Football players have very distinctive interests that are different from random accounts. They generally talk about the same latent topic (sport), thus the best results are obtained by the topic-based method.

Table 4.11: *Sizes of UCF Players Datasets*

	Number of users	**Number of Tweets**
Baseball players	62	5727
Football players	129	12500

Table 4.12: *Comparison of precision10 using NNP and Topic features in the sport domain: Baseball and Football players*

Domain	**Feature**	
	NNP	Topic
Baseball players	0.29	0.76
Football players	0.12	0.76

4.1.6 Related Work

Community detection is a fundamental task in social network analysis [132]. In the following we describe related work by considering methods that use links, semantics and content.

Network Clustering

The majority of approaches that performs community detection use social links (followers, retweets and user mentions) in order to detect communities as clusters of strongly (or densely) connected subgraphs [229,316]. Community detection in large graphs is a wide research topic, applied to many domains such as sociology, biology and finance. The methods used to detect community structures in graphs are based on modularity optimization [31], agglomerative clustering, centrality based and clique percolation [117]. In [187] the authors compared a multitude of community discovery algorithms, and computed the trade-offs between clustering objectives and community compactness.

All methods taken into account are computationally expensive in data acquisition, because in order to reconstruct significant sub-graphs it is necessary to query the Twitter API many times. Moreover, they cannot investigate the similarity of users who are not linked by social links. We cannot compare our results with these network-based approaches since our method does not require that users are socially connected. The networks of the datasets investigated in this section could even have no edges at all, resulting in meaningless networks measures, such as modularity [217].

The authors of [272, 273] tackle Influence Maximization task by including topic information to traditional information diffusion models on networks with a similar approach.

Semantic Methods

Another class of approaches uses the semantic content of social graphs to discover communities. In [257] the authors introduce a measure of signal strength between two nodes in the social network by using content similarity, while the authors of [324] propose the CUT (Community-User-Topic) model for discovering communities using the semantic content of the social graph. Communities are modeled as random mixtures over users who in turn have a topical distribution (interest) associated with them.

Other works use generative probabilistic modeling which considers both contents and links as being dependent on one or more latent variables, and then estimates the conditional distributions to find community assignments. Examples include PLSA-PHITS [69], Community-User-Topic model [324] and Link-PLSA-LDA [215]. For instance, Link-PLSA-LDA finds latent topics in text and citations and assumes different generative processes on citing documents, cited documents as well as citations themselves. Text generation follows the LDA approach, and link creation between citing and cited documents is controlled by topic-specific multinomial distributions.

In these approaches, content similarity between users plays a fundamental role, thereby underlining the relevance of content in community detection. These approaches have the same drawbacks in the data acquisition cost that was reported above.

Content-based Methods

Other works are more similar to our approach, as they use textual similarity, without deep semantic analysis. In [271] the authors proposes a method to cluster people in Twitter using words, by proposing a metric to weight the words; in [207] a method for computing user similarity is proposed, based on a network representing the semantic relationship between the words occurring in the same tweet and the related topic. Other methods discover user similarities based on content similarities; the method presented in [134] uses a regression model. Compared to our approach, these methods require a lot of data for building an accurate model of the terms used by Twitter accounts and are more focused on similarity discovery rather than community detection.

4.1.7 Conclusions

This study provides a systematic approach to user identification and community characterization in Twitter. We characterize syntactic and semantic features in tweets and then show which ones are most suited for testing community membership and cohesiveness. Proper nouns or latent content topics perform very well if used with cosine distance or Kullback-Leibler Divergence.

In several application contexts, our method achieves a precision@10 which is 70% or above (in our designed experiment, this means that only three accounts are incorrect out of ten, extracted from a total of 190 candidates, mediated over 50 executions). This result is remarkable if one considers that the proposed method is low-cost: it requires the extraction of the tweets of a candidate (through a single call to the standard Twitter API) and simple scripts, which internally calls standard libraries, for extracting from this corpus the frequencies of either topics or proper nouns; as we opted for a low-cost strategy, we preferred topics to instances as representative semantic features.

Our applications show one case (politics) where syntactic features (NNP) prevail over semantic ones (Topic) but also one case (targeted advertising for sports players) where the semantic features prevail over the syntactic ones. Moreover, the topic components and the features from nouns hint at the typical terms used within the community, thereby providing an interesting characterization of the community from a sociological perspective.

As input, the described method requires only a few examples of reference accounts considered similar by a domain expert, e.g., chess players or writers, to construct a sufficiently characterizing vocabulary. Keeping the size of the input low was one of the design goals of our work (to keep a manual search task manageable). However, we have also verified that the approach is robust to larger input sizes, as shown in Tables 4.1, 4.5 and 4.11. I tested datasets of different magnitudes that belong to communities with low and high specialized vocabularies, and the performances are comparable.

The practical implication of our study is in the extraction of targeted communities where each new candidate brings potentially high value. Targeted advertising, as discussed in Section 4.1.5, applies to many contexts. For example, under elections, it can be used by candidates who want to advertise just to potential voters of their party.

In this section, I reported a work comparing semantic and syntactic features to characterize communities. When communities debate about a selected topic, we talk about controversies. The following section describes a study investigating how to quantify the controversiality of topics by looking at the content shared by users on Twitter.

4.2 Measuring Controversy in Social Networks through NLP[1]

4.2.1 Introduction

Controversy in social networks is a phenomenon with an high social and political impact. Interesting analysis have been performed about presidential elections [277], congress decisions [136], hate spread [55], and harassing [176]. This phenomenon has been broadly studied from the perspective of different disciplines, ranging from the seminal analysis of conflicts within the members of a karate club [321] to political issues in modern times [1,3,74,89,212].

The irruption of digital social networks [107] gave raise to new ways of intentional intervention for taking advantages [55,277]. Moreover, highly contrasting points of view in groups tend to provoke conflicts that lead to attacks from one community to the other, such as harassing, “brigading", or “trolling" [176]. The existing literature reports a huge number of issues related to controversy, ranging from the splitting of communities and the biased information spread, to the increase of hate speeches and attacks between groups. For example, Kumar, Srijan, et al. [176] analyze many defense techniques from attacks on *Reddit* [9] while Stewart, et al. [277] insinuate that there was external interference in *Twitter* during the 2016 US presidential elections to benefit one candidate.

As shown in [175,179], detecting controversies also provides the basis to improve the *“news diet"* of readers, offering the possibility to connect users with different points of view by recommending them personalized content to read [213]. Other studies on “bridging echo chambers" [124] and the positive effects of inter group dialogue [6,232] suggest that direct engagement is effective for mitigating conflicts.

An accurate and automatic classifier of controversial topics, therefore, helps to develop quick strategies to prevent miss-information, fights and biases. Moreover, the identification of the main viewpoints and the detection of semantically closer users is also useful to lead people to healthier discussions. *Measuring* controversy is even more powerful, as it can be used to establish controversy levels. For this purpose, we propose a content-based pipeline to measure controversy on social networks, collecting posts' content about a fixed topic (an hashtag or a keyword) as root input.

Controversy quantification through vocabulary analysis also opens several research avenues, such as the analysis whether polarization is being created, maintained or aug-

[1] Authors: Juan Manuel Ortiz de Zarate, first author, conceptualization, data collection, implementation, experiment design, writing; Marco Di Giovanni, first author, conceptualization, data collection, implementation, experiment design, writing; Esteban Feuerstein, critical feedbacks, editing; Marco Brambilla, critical feedbacks, editing. [84] Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature; Springer eBook; Measuring Controversy in Social Networks Through NLP; Juan Manuel Ortiz de Zarate, Marco Di Giovanni, Esteban Zindel Feuerstein et al ©2020, https://doi.org/10.1007/978-3-030-59212-7_14

[9] https://www.reddit.com/

mented through community's way of talking.

Our main contribution can be summarized as the design of a controversy detection pipeline and the its application to 30 heterogeneous Twitter datasets. We outperform the state-of-the-art approaches, both in terms of accuracy and computational speed.

Our method is tested on datasets from Twitter. This microblogging platform has been widely used to analyze discussions and polarization [212,243,291,308,320]. It is a natural choice for this task, as it represents one of the main fora for public debate [308], it is a common destination for affiliative expressions [154] and it is often used to report and read news about current events [267]. An extra advantage is the availability of real-time data generated by millions of users. Other social media platforms offer similar data-sharing services, but few can match the amount of data and the documentation provided by Twitter. One last asset of Twitter for our work is given by *retweets* (sharing a tweet created by a different user), that typically indicate endorsement [24] and hence they help to model discussions as they can signal "who is with who".

This section is organized as follows: in Section 4.2.2 we list and summarize other works about controversy and polarization on social networks, in Section 4.2.3 we present the datasets collected for this study, while Section 4.2.4 contains the step-by-step description of our pipeline. In Section 4.2.5 we show the results and we conclude with Section 4.2.6.

4.2.2 Related Work

Due to its high social importance, many works focus on polarization measures in online social networks and social media [2,74,80,125,142]. The main characteristic that connects these works is that the measures proposed are based on the structural characteristics of the underlying social-graph. Among them, we highlight the work of Garimella et al. [125] that presents an extensive comparison of controversy measures, different graph-building approaches and data sources, achieving a state-of-the-art performance. We use this approach as a baseline to compare our results.

Matakos et al. [200] also develop a *polarization index* with a graph-based approach, not including text related features, modelling opinions as real numbers. Their measure successfully captures the tendency of opinions to concentrate in network communities, creating echo-chambers.

Other recent works [180,247,290] prove that communities may express themselves with different terms or ways of speaking, and use different jargon, which can be detected with the use of text-related techniques. Already described in Section 4.1, we also built very efficient classifiers and predictors of account membership within a given community by inspecting the vocabulary used in tweets, for many heterogeneous Twitter communities, such as chess players, fashion designers and members and supporters of political parties. In [290] Tran et al. found that language style, characterized using a hybrid word and part-of-speech tag n-gram language model, is a better indicator of community identity than topic, even for communities organized around specific topics. Finally, Lahoti et al. [180] model the problem of learning the liberal-conservative ideology space of social media users and media sources as a constrained non-negative matrix-factorization problem. They validate their model and solution on a real-world Twitter dataset consisting of controversial topics, and show that they are able to separate users by ideology with over 90% purity.

Other works for controversy detection through content have been performed over Wikipedia [101, 160] showing that text contents are good indicatives to estimate polarization. These works are heavily dependent on Wikipedia and can not be extrapolated to social networks.

In [159], the author explains controversy via generating a summary of two conflicting stances that build the controversy. Her work shows that a specific sub-set of tweets is enough to represent the two opposite positions in a polarized debate.

Quantifying Controversy on Social Media

The study described in this chapter is largely inspired by the work of Garimella et al. [125], summarized here.

This is the first large-scale systematic study for quantifying controversy in social media, Twitter in particular. While most of the previous works were topic-dependent, usually case-studies of important events (i.e. political elections), their approach can be applied to any topic with a large enough volume of discussion on social media.

They design a three-stages pipeline that not only identifies a controversy, but also quantifies the degree of controversy. The approach is graph-based but includes a short description of a couple of content-based methods that they tried without success.

Firstly, defining a topic of discussion on social media is not straightforward. They define it as a set of hashtags, since controversial topics usually includes tweets containing different hashtags for different sides of the controversy. This set is built starting from a seed hashtag and evaluating candidates' hashtags (hashtags co-occurring to the seed) using a similarity score specifically designed to handle also very popular hashtags (e.g. #follow).

Then, they collect all the tweets using one or more of those hashtags during a fixed time window, to obtain the final dataset to analyze. A conversation graph is built where nodes are users and edges can be retweets (retweet graph), follows (follow graph) or shared content (content graph).

METIS [168] algorithm is used to split the obtained graphs in two partitions and the graph layout is produced by Gephi's ForceAtals2 [158] algorithm. The graph layout is qualitatively evaluated to check if the intuition that controversial topics produce polarized graphs is correct.

Finally, many controversy measures are defined, computed and compared. The best one is based on random walks on the graph. The final score is a comparison between walks that start and end in the same partition and those that cross the sides. Other original measures tried includes betweenness centrality measures and embedding computed by ForceAtlas2 compared calculating distances, while selected baselines involve the definition of boundaries of the graph partitions and an application of dipole moment.

Their pipeline is tested on 20 Twitter datasets resulting that retweet graph and RWC is the best approach. Their analysis includes the application of the same pipeline also to external datasets, an analysis of controversy during time for a specific event and the application on synthetic graphs.

Finally, they try two content-based methods obtaining unsatisfying results. The first one is based on Bag-of-Words (BoW, Section 2.2.1) representations of strings. Tweets are cleaned and vectorized and then clustered using CLUTO algorithm with cosine distance. Then, they apply KL distance on the two vocabularies and I2 measure of clusters heterogeneity. They were unable to reject the null hypothesis with $p = 0.05$, thus they conclude that the two sides of the controversy use a similar vocabulary.

The second method involves sentiment analysis. Applying SentiStrength, an algorithm to quantify the sentiment of a text, on tweets, they obtain a distribution of values in $[-4, 4]$. While the average values of score for controversial and not controversial topics are similar, the variances are very different. Controversial topics have much different tones (higher variance) than non-controversial ones. However, this approach is extremely language-dependent and thus not compared to the other methods.

Vocabulary-based Method for Quantifying Controversy in Social Media

Mainly based on the previous work of Garimella, this work [85] uses advances in NLP to quantify controversy in social media through content, not only proving that in texts there is enough information to capture the level of controversy in a discussion, but also finding a faster approach with similar accuracies.

Their definition of topic is similar to the previous one, not involving a set of hashtags but substituting it to a general keyword, when different hashtags are related to sides of controversy.

Their pipeline also starts with a building retweet-graph phase, considering only one retweet as enough to establish connection instead of two. Then, the graph partition phase is performed using Louvain or Walktrap methods, allowing to not fix a priori the number of clusters, thus handling better cases when there are more than two opposite sides of a controversy.

Then, tweets of users belonging to one of the two biggest clusters are collected, cleaned and merged to create a training dataset for FastText (Section 2.3.2). The ground truth used are the labels obtained by the graph partitioning algorithm. A subset of users is selected looking at prediction made by FastText. Only users obtaining a score higher than 0.9 are selected, removing noise of uncertain users. Finally, a controversy score is computed using the dipole moment equations, already introduced in [212], preceded by a label-propagation step.

The designed pipeline outperforms the state-of-the-art approach of Garimella et al. [125] in terms of computational time, obtaining similar results in ROC AUC scores.

This section describes an even more complete and detailed work about quantifying controversy on social media using tweets content. The main difference between this work and [85] is that the techniques presented here are less dependent on the graph structure. Our new content-based pipeline introduces the possibility of defining and detecting concepts like the "semantic frontier" of a cluster. This opens new ways to activate interventions in the communities, such as the investigation of users lying near that frontier to facilitate a healthier interaction between the communities, or the analysis of users far away from the frontier to understand which aspects establish the real differences. Improvements on [85] (used as a second baseline in this work), include a wider comparison of NLP models and distance measures, a higher heterogeneity of datasets used, and results in better performances both in terms of AUC ROC scores and computational times.

4.2.3 Datasets

To test our approach, we collect 30 Twitter datasets in six languages. Each dataset corresponds to a manually selected topic among the trending ones. The collection is performed through the official Twitter API[10].

Topic definition

In the literature, a topic is often defined by a single hashtag. We believe that this might be too restrictive since some discussions may not have a defined hashtag, but they are about a *keyword* that represents the main concept, i.e. a word or expression that is not specifically an hashtag but it is widely used in the discussion. For example during the Brazilian presidential elections in 2018, we collected tweets mentioning to the word *Bolsonaro*, the principal candidate's surname. Thus, in our approach, a topic is defined as a specific hashtag or keyword, depending on the discussion. For each topic we collect all the tweets that contain its hashtag or keyword, posted during a selected observation window. We also check that each topic is associated with a large enough activity volume.

[10] https://developer.twitter.com/en/docs

Description of the datasets

We collected 30 discussions (50% more than the baseline work [125]) that took place between 2015 and 2020, half of them controversial and half not. We selected discussions in six languages: English, Portuguese, Spanish, French, Korean and Arabic, occurring in five regions over the world: South and North America, Western Europe, Central and Southern Asia. The details of each discussion are described in Table A.1. We have chosen discussions clearly recognizable as controversial or not to have an evident ground truth. Blurry discussions will be analyzed in future works. The encoded datasets are available on github[11].

Since our models require a large amount of text and since a tweet contains no more than 240 characters, we established a threshold of at least 100000 tweets per topic. Topics containing a lower number of tweets were discarded. To select discussions and to determine if they are controversial or not we looked for topics widely covered by mainstream media that have generated ample discussion, both online and offline. For non-controversial discussions we focused both on “soft news" and entertainment, and on events that, while being impactful and/or dramatic, did not generate large controversies. On the other side, for controversial debates we focused on political events such as elections, corruption cases or justice decisions. We validate our intuition by manually checking random samples of tweets.

To furtherly establish the presence or absence of controversy of our datasets, we visualized the corresponding networks through ForceAtlas2 [158], a widely used force-directed layout. This algorithm has been recently found to be very useful at visualizing community interactions [300], as it represents closer users interacting among each other, and farther users interacting less. Figure 4.1 shows examples of how non-controversial and controversial discussions respectively look like with ForceAtlas2 layout. As we can see in these figures, in a controversial discussion the layout shows two well separated groups, while in a non-controversial one it generates one big cluster.

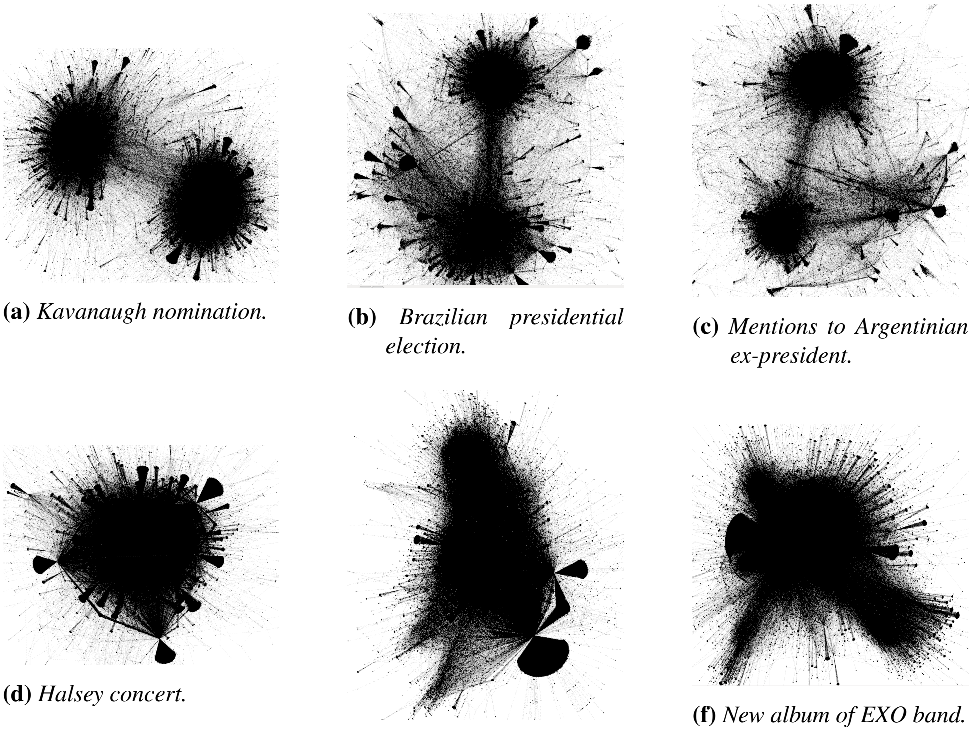
More information on the datasets is given in Table A.1 in Appendix A.

4.2.4 Methodology

Our approach can be outlined into four phases, namely *graph building* phase, *community identification* phase, *embedding* phase and *controversy score computation* phase. The final output of the pipeline is a positive value that measures the controversy of a topic, with higher values corresponding to lower degrees of controversy.

Our hypothesis is that using the embeddings generated by an NLP model, we can distinguish different ways of speaking; the more controversial the discussion is, the better differentiation we obtain.

[11] Code and datasets used in this work are available here: https://github.com/jmanuoz/Measuring-controversy-in-Social-Networks-through-NLP



(a) *Kavanaugh nomination.*

(b) *Brazilian presidential election.*

(c) *Mentions to Argentinian ex-president.*

(d) *Halsey concert.*

(e) *Pop star birthday.*

(f) *New album of EXO band.*

Figure 4.1: *ForceAtlas2 layout for different discussions. (a), (b) and (c) are controversial while (d), (e) and (f) are non-controversial.*

Graph Building Phase

Firstly, our purpose is to build a conversation graph that represents activities related to a single topic of discussion. For each topic, we build a retweet-graph G where each user is represented by a vertex, and a directed edge from node u to node v indicates that user u retweeted a tweet posted by user v.

We selected to build a retweet-graph because retweets usually indicate endorsement [24] and are not constrained to “follower" links. Users retweets to propagate to their followers an opinion that they share expressed by an user that they follow or not.

As typically in the literature [27,55,114,177,212,277] we establish that one retweet among a pair of users is enough to define an edge between them. We do not use “quotes" to build the graph since, due to their nature, they can both signal endorsement and opposition, allowing users to comment the quoted tweet.

We remark that the “retweet information" is included in the tweets extracted, allowing us to build the graph without increasing the number of twitter API requests needed. This makes this stage faster than, for example, building a follower graph, another popular alternative.

Community Identification Phase

To identify the jargon of the community we need to be very accurate at defining its members. If we, in our will of finding two principal communities, force the partition of the graph in that precise number of communities, we may be adding noise in the jargon of the principal communities that are fighting each other. Thus, we decide to cluster the graph using Louvain [31], one of the most popular graph-clustering algorithms. It is a greedy technique that can run over big networks without memory or running time problems, and does not detect a fixed number of clusters. Its output depends on the Modularity Q optimization, resulting in more or less “noisy" communities. In a polarized context there are two principal sides covering the whole discussion, thus we take the two biggest communities identified by Louvain and use them for the following steps. Since, to have controversy in a discussion, there must be “at least" two sides, if the principal sides are more than two, discarding the smallest ones will not impact the final result. Up to here the approach we follow is the same as in [85].

Embedding Phase

In this phase, our purpose is to embed each user into a corresponding vector. These vectors encode syntactic and semantic proprieties of the posts of the corresponding accounts. They will be used in the next phase to compute the controversy score, since we need fixed dimension semantically significant vectors to perform the following computations.

Firstly, tweets belonging to the users of the two principal communities selected in the previous stage are grouped by user and cleaned. We remove duplicates and, from each tweet, we remove usernames, links, punctuation, tabs, leading and lagging blanks, general spaces and the retweet keyword “RT", the string that points that a tweet is a retweet. Many sentence embedding techniques have been developed in the literature, ranging from simple BoW models to complex LMs. To perform this step we selected two models:

- FastText (Section 2.3.2). The hyperparameters are defined using the findings of [319] on Twitter data. We train this model with tagged data, accordingly to the output of Louvain (previous stage), representing the community of the user and we use the trained model to compute the text embedding.
- BERT base (Section 2.4.4), that embeds texts into fixed dimension vectors encoding semantically significance and meaning. We finetune BERT on a 2-classes classification task for 6 epochs (learning rate set to 10^{-5}) on the dataset previously described. Since our goal is to obtain embeddings of tweets, after the training procedure we remove the fully-connected layer and we use the outputs of BERT as embeddings. In detail, BERT firstly split a sentence into tokens, adding the *[CLS]*

token at the beginning and the *[SEP]* at the end. Then, it embeds each token into a 786-dimensional vector. Since we need a single vector of fixed length to compute our score, we select as aggregator the embedding of the *[CLS]* token. This is the same strategy selected during the fine-tuning step. We perform this stage using bert-as-service [314].

To train FastText and BERT in a supervised way, we need to create a labeled training set. We label each user with its community, namely with tags C_1 and C_2, corresponding respectively to the biggest (Community 1) and second biggest (Community 2) groups. It is important to note that, to prevent bias in the model, we take the same number of users from each community, downsampling the first principal community to the number of users of the second one.

Controversy Score Computation Phase

To compute the controversy score, we select some users as the best representatives of each side's main point of view. We run the HITS algorithm [171] to estimate the authoritative and hub score of each user. We take the 30% of the users with the highest authoritative score and the 30% with the highest hub score and we call them *central users*.

Finally, we compute the controversy score r, using the embeddings of the central users $x_i \in \mathbb{R}^k$ and the labels $y_i \in \{1, 2\}$, imposing their belonging to cluster C_1 or C_2, computed during the community identification phase.

We compute the centroids of each cluster j with equation 4.1, where $|C_j|$ is the magnitude of cluster C_j, and a global centroid c_{glob} with equation 4.2.

$$c_j = \frac{1}{|C_j|} \sum_{i:y_i=j} x_i \tag{4.1}$$

$$c_{glob} = \frac{1}{|C_1| + |C_2|} \sum_{i} x_i \tag{4.2}$$

We define D_j as the sum of distances between the embeddings x_i and their centroids c_j using equation 4.3 for $j = 1, 2$, where $dist$ is a generic distance function. Similarly, D_{glob} is the sum of distances between all the embeddings and the global centroid.

$$D_j = \sum_{i:y_i=j} dist(x_i, c_j) \tag{4.3}$$

Because of the *curse of dimensionality* [21], measuring distances over big number of dimensions is not a trivial task and the usefulness of a distance measure depends on the sub-spaces that the problem belongs to [261]. For this reason, we select and test four distance measure: L_1 (Manhattan), L_2 (Euclidean), Cosine and Mahalanobis [83] distance (particularly useful when the embedding space is not interpretable and not homogeneous, since it takes into account also correlations of the dataset and reduces to Euclidean distance if the covariance matrix is the identity matrix).

The controversy score r is defined in equation 4.4.

$$r = \frac{D_1 + D_2}{D_{glob}} \tag{4.4}$$

Intuitively, it represents how much the clusters are separated. We expect that, if the dataset is a single cloud of points, this value should be near 1 since the two centroids c_1 and c_2 will be near each other and near the global centroid c_{glob}. On the contrary, if the embeddings successfully divide the dataset in two clearly separated clusters, their centroids will be far apart and near to the points that belong to their own clusters. Note that r is, by definition, positive, since D_1, D_2 and D_{glob} are positive too.

The datasets and the full code is available on github[12] and the results discussed in the following section are fully reproducible.

4.2.5 Results

In this section we collect the results obtained with the different techniques described above and we compare them to the state-of-the-art structured-based method “RW" [125] and the previous work “DMC" [85], a structure and text-based approach. In Figure 4.2 we show the distributions of scores of FastText and BERT, using the four different distances described before, compared to the baselines “RW" and “DMC". We plot them as beanplots with scores of controversial datasets on the left side and non-controversial ones on the right side. Note that, since by definition “DMC" approach gives higher scores for controversial datasets and lower scores for non-controversial ones, the two distributions are reversed.

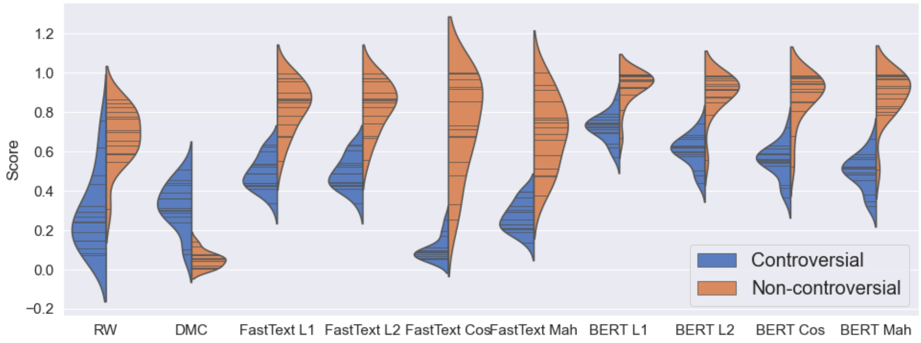


Figure 4.2: *Scores distributions comparison.*

[12] https://github.com/jmanuoz/Measuring-controversy-in-Social-Networks-through-NLP

The less the two distributions overlap, the better the pipeline works. Thus, to quantify the performance of different approaches, we compute the ROC AUC. By definition, this value is between 0 and 1, where 0.5 means that the curves are perfectly overlapped (i.e. random scoring), while values of 0 and 1 correspond to perfectly separated distributions. The comparison among the different distance measures is reported in Table 4.13. The best score (the highest value) is obtained by FastText model with cosine distance, outperforming the state-of-the-art methods [85, 125].

Table 4.13: *ROC AUC scores comparison*

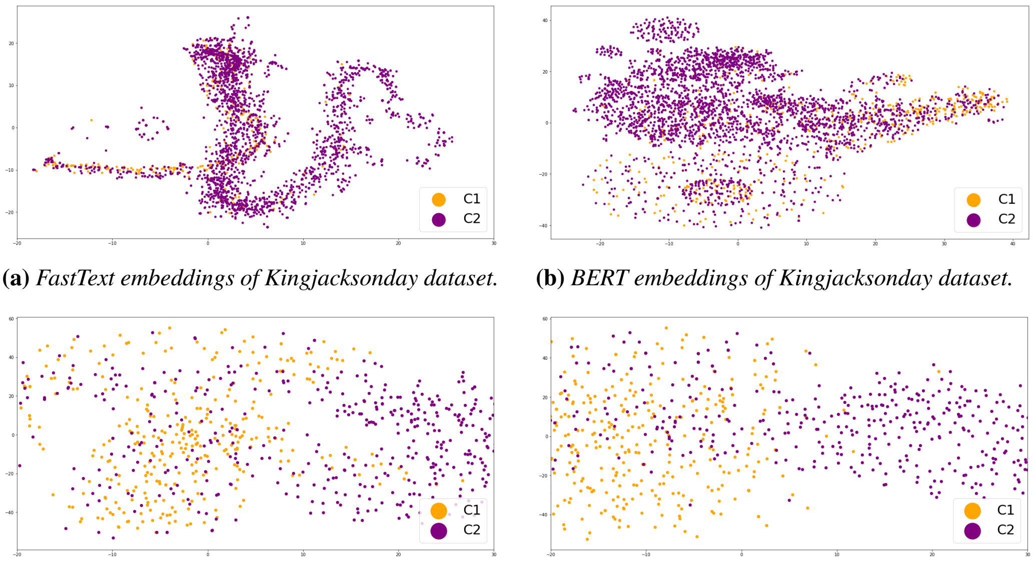
Method	L1	L2	Cosine	Mahalanobis	Baseline
FastText	0.987	0.987	**0.996**	0.991	-
BERT	0.942	0.947	0.942	**0.964**	-
DMC	-	-	-	-	**0.982**
RW	-	-	-	-	**0.924**

Although BERT reached many state-of-the-art results in different NLP tasks [91], FastText suits better in our pipeline. Analyzing the wrongly scored cases we observe that BERT fails mainly with the non-controversial datasets, for example *Feliz Natal* dataset (0.51 controversy score). Our hypothesis is that, since BERT is a bigger and more complex model than FastText, sometimes it overfits the data. BERT is able to separate the two communities' ways of speaking even when they are very similar, not opposite sides of a controversy, exploiting differences that we are not able to perceive.

To qualitatively check this behavior we plot the embeddings produced by each technique by reducing their dimension to 2 with t-SNE algorithm [296] for visualization purposes. In Figure 4.3 we show the reduced embeddings obtained by each method for two non-controversial datasets *Jackson's birthday* and *Feliz Natal*. The first dataset is correctly predicted as non-controversial by both methods and we can see that their embeddings are highly mixed, as expected. However *Feliz Natal* embeddings are mixed when FastText is used, while BERT is still able to split them in two separate clusters. This shows that, for the *Feliz Natal* case, BERT is still differentiating two ways of speaking.

Computational Time

Figure 4.4 shows the boxplots over the 30 datasets of the total computational times (in seconds) of our two best algorithms, from the beginning (graph building stage) to the end (controversy score computation stage), compared to the baselines. Our approaches are faster than the baseline graph-based method (RW), while DMC approach is only faster than our BERT variant. Fastext approach outperforms both the baselines, allowing a quicker analysis when used in a real-time perspective, since intervention could be necessary for prevention of malicious behaviors, already described in Section 4.2.1.



(a) *FastText embeddings of Kingjacksonday dataset.* **(b)** *BERT embeddings of Kingjacksonday dataset.*

(c) *FastText embeddings of Feliz Natal dataset.* **(d)** *BERT embeddings of Feliz Natal dataset.*

Figure 4.3: *t-SNE reduced embeddings produced by Fasttex and Bert.*

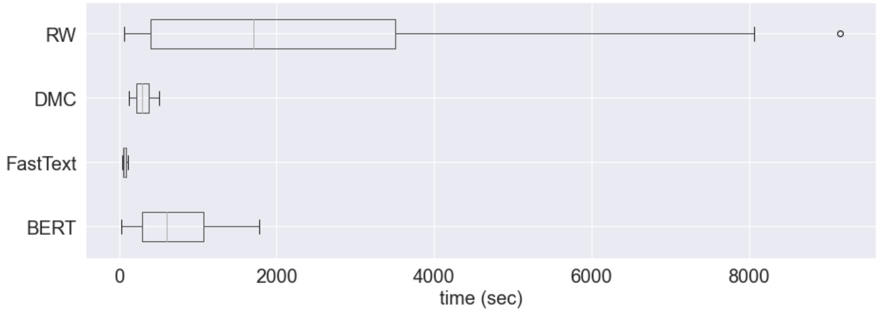


Figure 4.4: *Computational time comparison.*

4.2.6 Conclusions

In this work, we designed an NLP-based pipeline to measure controversy. We tested variants, such as two embedding techniques (using FastText and BERT language models) and four distance measures. We applied these approaches on 30 heterogeneous Twitter datasets, and we compared the results. THe best variant, using FastText and cosine distance, outperforms not only the state-of-the-art graph-based method [125], where the authors state that content-based techniques do not perform as well as structure-based ones, but also the previous work [85], in terms of ROC AUC score and speed, due to the lower dependence on the graph structure.

This pipeline involves FastText, a fast model to encode sentences, or BERT, a more accurate language model, slower due to the complex fine-tuning process required. FastText obtains the best performance overall, reaching a ROC AUC score of 0.996. As we previously reported, BERT is so strong that it could differentiate ways of speaking even when they are not in conflict. Due to the nature of the pipeline, FastText performs better and requires less computing time. These results are the first steps towards helping people to participate in healthier discussions.

Since this approach on controversy detection shares some similarities with previous works, we share some limitations too:

- *Evaluation*: difficulty to establish the ground-truth since the definition of controversial topics is sometimes debatable;
- *Multisided controversies*: the approach cannot detect and quantify controversies with more than two sides;
- *Choice of data*: the manual collection of topics could bias the results;
- *Overfitting*: the number of datasets is small even if now we have ten more discussions than previous works.

This language-based approach has other limitations. Firstly, training accurate NLP models requires a significant amount of text, limiting the pipeline to trending discussions. However, the most intriguing controversies have consequences at a societal level. Secondly, we built the approach on Twitter. Twitter is one of the most used social networks for an online discussion, and it is relatively easy to collect the required data. However, Twitter's characteristic limit of 280 characters per message (140 till a short time ago) is an intrinsic limitation. I plan to apply this pipeline to other social networks like Facebook or Reddit.

In this section, I observed that a simple approach (FastText) outperforms a deep Language Model (BERT) for controversiality quantification of topics. However, the effectiveness of Transformer-based models on many heterogeneous tasks suggests that a straightforward application of the model is not always sufficient to obtain good results. In the next section, I describe an approach to train a Transformer-based model on Semantic Sentence Embeddings using large corpora of texts from Twitter. This model maps texts to dense high-dimensional vectors that encode crucial information to detect semantic similarities. Thus in the last section of this chapter, I apply it to model users and, finally, to detect communities and controversies more accurately, exploiting the full power of deep Language Models.

4.3 Exploiting Twitter as Source of Large Corpora of Weakly Similar Pairs for Semantic Sentence Embeddings[1]

4.3.1 Introduction and Related Work

Word-level embeddings techniques compute fixed-size vectors encoding semantics of words [204, 230], usually unsupervisedly trained from large textual corpora. It has always been more challenging to build high-quality sentences-level embeddings.

Currently, sentence-embeddings approaches are supervisedly trained using large labeled datasets [58, 60, 72, 103, 157, 252, 310], such as NLI datasets [41, 312] or paraphrase corpora [99]. Round-trip translation has been also exploited, where semantically similar pairs of sentences are generated translating the non-English side of NMT pairs, as in ParaNMT [309] and Opusparcus [76]. However, large labeled datasets are rare and hard to collect, especially for non-English languages, due to the cost of manual labels, and there exists no convincing argument for why datasets from these tasks are preferred over other datasets [56], even if their effectiveness on STS tasks is largely empirically tested.

Therefore, recent works focus on unsupervised approaches [56, 130, 188, 194, 304], where unlabeled datasets are exploited to increase the performance of models. These works use classical formal corpora such as OpenWebText [135], English Wikipedia, obtained through Wikiextractor [14], or target datasets without labels, such as the previously mentioned NLI corpora.

Instead, we propose a Twitter-based approach to collect large amounts of *weak* parallel data: the obtained couples are not exact paraphrases like previously listed datasets, yet they encode an intrinsic powerful signal of relatedness. We test pairs of quote and quoted tweets, pairs of tweet and reply, pairs of co-quotes and pairs of co-replies. We hypothesize that quote and reply relationships are weak but useful links that can be exploited to supervisedly train a model generating high-quality sentence embeddings. This approach does not require manual annotation of texts and it can be expanded to other languages spoken on Twitter.

We train models using triplet-like structures on the collected datasets and we evaluate the results on the standard STS benchmark [57], two Twitter NLP datasets [182, 315] and four novel benchmarks.

Our contributions are four-fold: we design an language-independent approach to collect big corpora of weak parallel data from Twitter; we fine-tune Transformer based models with triplet-like structures; we test the models on semantic similarity tasks, including four novel benchmarks; we perform ablation on training dataset, loss function, pre-trained initialization, corpus size and batch size.

[1] Authors: Marco Di Giovanni: first author, conceptualization, data collection, implementation, experiment design, writing; Marco Brambilla: critical feedbacks, editing. [95]

4.3.2 Datasets

We download the general Twitter Stream collected by the Archive Team Twitter[13]. We select English[14] tweets posted in November and December 2020, the two most recent complete months up to now. They amount to about $27G$ of compressed data ($\sim 75M$ tweets).[15] This temporal selection could introduce biases in the trained models since conversations on Twitter are highly related to daily events. We leave as future work the quantification and investigation of possible biases connected to the width of the temporal window, but we expect that a bigger window corresponds to a lower bias, thus a better overall performance.

We collect four training datasets: the Quote Dataset, the Reply Dataset, the Co-quote Dataset and the Co-reply Dataset.

- The **Quote Dataset (Qt)** is the collection of all pairs of quotes and quoted tweets. A user can *quote* a tweet by sharing it with a new comment (without the new comment, it is called *retweet*). A user can also retweet a quote, but it cannot quote a retweet, thus a quote refers to an original tweet, a quote, or a reply. We generate *positive* pairs of texts coupling the quoted texts with their quotes;

- The **Reply Dataset (Rp)** is the collection of all couples of replies and replied tweets. A user can reply to a tweet by posting a public comment under the tweet. A user can reply to tweets, quotes and other replies. It can retweet a reply, but it cannot reply to a retweet, as this will be automatically considered a reply to the original retweeted tweet. We generate *positive* pairs of texts coupling tweets with their replies;

- The **Co-quote Dataset (CoQt)** and **Co-reply Dataset (CoRp)** are generated respectively from the Qt Dataset and the Rp Dataset, selecting as *positive* pairs two quotes/replies of the same tweet.

To avoid *popularity-bias* we collect only one positive pair for each quoted/replied tweet in every dataset, otherwise viral tweets would have been over-represented in the corpora.

We clean tweets by lowercasing the text, removing URLs and mentions, standardizing spaces and removing tweets shorter than 20 characters to minimize generic texts (e.g., variations of "Congrats" are common replies, thus they can be usually associated to multiple original tweets). We randomly sample $250k$ positive pairs to train the models for each experiment, unless specified differently, to fairly compare the perfor-

[13] https://archive.org/details/twitterstream

[14] *English* tweets have been filtered accordingly to the “lang" field provided by Twitter.

[15] We do not use the official Twitter API because it does not not guarantee a reproducible collections (Tweets and accounts are continuously removed or hidden due to Twitter policy or users’ privacy settings).

mances (in Section 4.3.5 we investigate how the corpus size influences the results). We also train a model on the combination of all datasets (**all**), thus $1M$ text pairs.

We show examples of pairs of texts from the four datasets in Figure A.1 and A.2 in Appendix A.

4.3.3 Approach

We select triplet-like approaches to train a Tranformer model on our datasets. We extensively implement our models and experiments using sentence-transformers python library[16] and Huggingface [313]. Although the approach is model-independent, we select four Transfomer models [298] as pre-trained initializations, currently being the most promising technique ($\sim 110M$ parameters):

- **RoBERTa base** [192] is an improved pre-training of BERT-base architecture [91], to which we add a pooling operation: MEAN of tokens of last layer. Preliminary experiments of pooling operations, such as MAX and [CLS] token, obtained worse results;
- **BERTweet base** [218] is a BERT-base model pre-trained using the same approach as RoBERTa on $850M$ English Tweets, outperforming previous SOTA on Tweet NLP tasks, to which we add a pooling operation: MEAN of tokens of last layer;
- **Sentence BERT** [252] models are BERT-base models trained with siamese or triplet approaches on NLI and STS data. We select two suggested base models from the full list of trained models: bert-base-nli-stsb-mean-tokens (S-BERT) and stsb-roberta-base (S-RoBERTa).

We test the two following loss functions:

- **Triplet Loss (TLoss)**: given three texts (an anchor a_i, a positive text p_i and a negative text n_i), we compute the text embeddings (s_a, s_p, s_n) with the same model and we minimize the following loss function:

  $$max(||s_a - s_p|| - ||s_a - s_n|| + \epsilon, 0)$$

  For each pair of anchor and positive, we select a *negative* text randomly picking a positive text of a different anchor (e.g., about the Quote dataset, anchors are quoted tweets, positive texts are quotes and the negative texts are quotes of different quoted tweets);
- **Multiple Negative Loss (MNLoss)** [151]: given a batch of positive pairs:

  $$(a_1, p_1), ..., (a_n, p_n)$$

[16] https://github.com/UKPLab/sentence-transformers

> we assume that (a_i, p_j) is a negative pair for $i \neq j$ (e.g., Quote Dataset: we assume that quotes cannot refer to any different quoted tweet). We minimize the negative log-likelihood for softmax normalized scores. We expect the performance to increase with increasing batch sizes, thus we set $n = 50$, being the highest that fits in memory (see Section 4.3.5 for more details).

We train the models for 1 epoch[17] with AdamW optimizer, learning rate 2×10^{-5}, linear scheduler with 10% warm-up steps on a single NVIDIA Tesla P100. Training on $250k$ pairs of texts requires about 1 hour, on $1M$ about 5 hours.

4.3.4 Evaluation

We evaluate the trained models on seven heterogeneous semantic textual similarity (STS) tasks: four novel benchmarks from Twitter, two well-known Twitter benchmarks and one classical STS task. We planned to test the models also on Twitter-based classification tasks, e.g., Tweeteval [18]. However, the embeddings obtained from our approach are *not* designed to transfer learning to other tasks, but they should mainly succeed on similarity tasks. A complete and detailed evaluation of our models on classification tasks is also not straightforward, since a classifier must be selected and trained on the top of our models, introducing further complexity to the study. We leave this analysis for future works.

Novel Twitter benchmarks

We propose four novel benchmarks from the previously collected data. Tweets in these datasets are discarded from *every* training set to avoid unfair comparisons. We frame these as ranking tasks and we pick normalized Discounted Cumulative Gain (nDCG) as metric [161][18]. We propose these datasets to highlight that benchmark approaches are not able to detect similarities between related tweets, while they can easily detect similarities between formal and accurately selected texts. Thus the necessity for our new models.

Direct Quotes/Replies (DQ/DR): Collections of $5k$ query tweets, each one paired with 5 positive candidates (quotes/replies of the query tweets) and 25 negative candidates (quotes/replies of other tweets). We rank candidates by cosine distance between their embeddings and the embedding of the query tweet.

Co-Quote/Reply (CQ/CR): Similar to previous tasks, we focus on co-quotes/co-replies, i.e., pairs of quotes/replies of the same tweet. These datasets are collections

[17] We briefly tested the training for two epochs in preliminary experiments, but we noticed no evident benefits. Moreover, increasing the number of epochs enhances the risk of overfitting the noise included in tweets since these texts are noisy and we do not perform validation.

[18] nDCG is a common ranking-quality metric obtained normalizing Discounted Cumulative Gain (DCG). The scores range from 0 to 1, the higher the better. Thus, 1 represents a perfect ranking: the first ranked document is the most relevant one, the second ranked document is the second most relevant one, and so on.

of $5k$ query quotes/replies, each one paired with 5 positive candidates (quotes/replies of the same tweet) and 25 negative candidates (quotes/replies of other tweets). We rank candidates by cosine distance between their embeddings and the embedding of the query tweet.

Established benchmarks

We select two benchmarks from Twitter, PIT dataset and Twitter URL dataset (TURL), and the STS benchmark of formal texts. We pick Pearson correlation coefficient (Pearson's r) as metric.

PIT-2015 dataset [315] is a Paraphrase Identification (PI) and Semantic Textual Similarity (SS) task for the Twitter data. It consists in 18762 sentence pairs annotated with a graded score between 0 (no relation) and 5 (semantic equivalence). We test the models on SS task.

Twitter URL dataset [182] is the largest human-labeled paraphrase corpus of 51524 sentence pairs and the first cross-domain benchmarking for automatic paraphrase identification. The data are collected by linking tweets through shared URLs, that are further labeled by human annotators, from 0 to 6.

STS benchmark datasets [57] is a classical dataset where pairs of formal texts are scored with labels from 0 to 5 as semantically similar. It has been widely used to train previous SOTA models, so we do not expect our models trained on informal weak pairs of texts to outperform them. However, it is a good indicator of the quality of embeddings and we do expect our models to not deteriorate on accuracy with respect to their initialized versions.

Baselines

We compare our models with the pre-trained initializations: RoBERTa and BERTweet (MEAN pooling of tokens) and S-BERT and S-RoBERTa, pre-trained also on STSb.

4.3.5 Results and Ablation Study

In Table 4.14 we show the results of the experiments.

As expected, we conclude that baseline models perform poorly in the new benchmarks, being trained for different objectives on different data, while *Our-BERTweet (all)* obtains the best performances. On established datasets, our training procedure improves the corresponding pre-trained versions. The only exception is when our model is initialized from S-BERT and S-RoBERTa and tested on TURL, where we notice a small deterioration of performances (0.5 and 0.1 points respectively) and on STSb-test, since baselines where trained on STSb-train. This result proves that our corpora of weakly similar texts are valuable training sets and specific NLI corpora are not nec-

Table 4.14: *$nDCG \times 100$ (novel benchmarks) and Pearson's $r \times 100$ (established benchmarks). We indicate our models with the Our- prefix followed by the name of the initialization model, between parentheses the training dataset. If not specified, we use MNLoss. Results are averages of 5 runs.*

Model	DQ	CQ	DR	CR	Avg	PIT	TURL	STSb
RoBERTa-base	42.9	39.1	55.0	41.0	44.5	39.5	49.7	52.5
BERTweet	46.9	42.5	56.7	44.1	47.5	38.5	48.2	48.2
S-BERT	53.7	43.9	60.5	45.4	50.9	43.8	**69.9**	84.2
S-RoBERTa	52.4	42.8	59.1	44.1	49.6	57.3	69.1	**84.4**
Our-RoBERTa-base (all)	80.8	68.5	83.0	66.1	74.6	58.8	67.5	74.2
Our-BERTweet (all)	83.7	72.1	84.2	68.3	**77.1**	66.1	67.1	72.4
Our-S-BERT (all)	79.0	66.6	81.5	64.6	72.9	57.7	69.4	76.1
Our-S-RoBERTa (all)	80.2	67.8	82.6	65.6	74.0	60.1	69.0	78.9
Our-RoBERTa (Qt)	75.9	63.6	79.3	61.2	70.0	60.7	66.8	74.9
Our-BERTweet (Qt)	80.8	68.9	81.7	65.0	74.1	**67.4**	66.0	72.4
Our-S-BERT (Qt)	73.6	61.5	77.7	59.8	68.1	57.6	69.1	79.3
Our-S-RoBERTa (Qt)	74.6	62.6	78.4	60.5	69.0	58.1	68.8	80.7
Our-BERTweet (Co-Qt)	80.7	70.6	80.8	65.9	74.5	63.6	64.3	70.9
Our-BERTweet (Rp)	81.5	68.4	82.2	65.8	74.5	63.8	67.3	72.3
Our-BERTweet (Co-Rp)	79.3	69.0	81.7	67.5	74.4	62.1	64.3	67.3
Our-BERTweet-TLoss (Qt)	67.7	60.8	71.5	56.9	64.2	53.1	43.4	44.7

essary to train accurate sentence embeddings. We remark that for many non-English languages, models such as S-BERT and S-RoBERTa cannot be trained since datasets such as STSb-train do not exist yet[19].

The best initialization for novel benchmarks and PIT is BERTweet, being previously unsupervisedly trained on big amounts of similar data, while for TURL and STSb the best initializations are S-BERT and S-RoBERTa respectively. MNLoss always produces better results than a simple TripletLoss, since the former compares multiple negative samples for each positive pair, instead of just one as in the latter.

The training dataset does not largely influence the performance of the model on novel benchmarks, while, on established benchmarks, Qt and Rp are usually better than CoQt and CoRp training datasets. However, the concatenation of all datasets (all) used as training set almost always produces better results than when a single dataset is used.

Figure 4.5 (left) shows that performances improve by increasing the corpus size of Qt dataset. Since they do not reach a plateau yet, we expect better performances when a wider magnitude of Tweets is collected.

Figure 4.5 (right) shows the performance of the same model when varying batch size in MNLoss, i.e., the number of negative samples for each query. The performance plateaus at about 10, setting a sufficient number of negative samples. However, we set it to a higher value because it implies a faster training step.

[19] Recently, multilingual approaches have been succesfully tested [253].

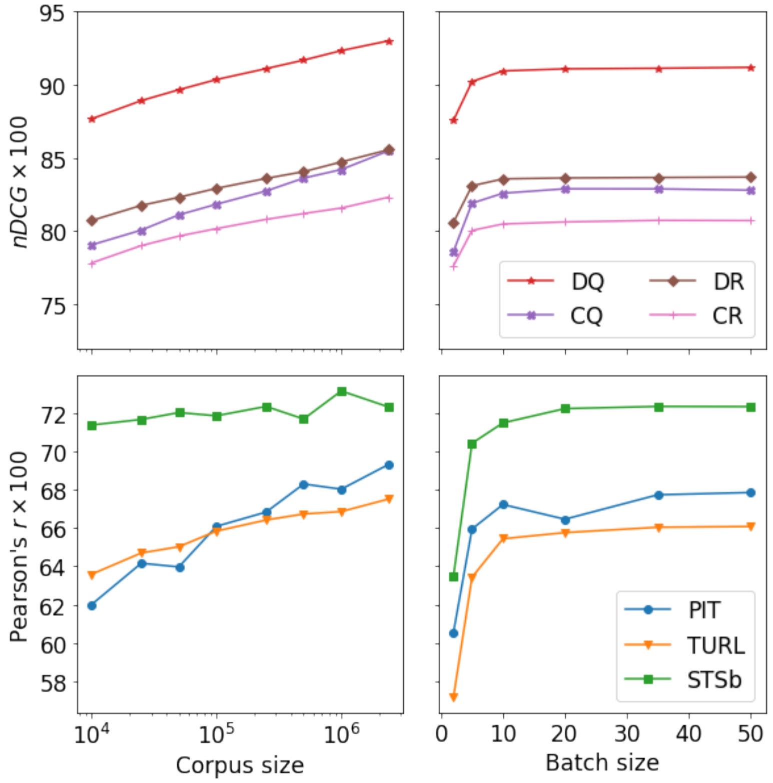


Figure 4.5: *$nDCG \times 100$ and Pearson's $r \times 100$ varying Corpus size (left) and Batch size (right) on Our-BERTweet trained on Quote dataset with MNLoss. Results are averages of 5 runs.*

4.3.6 Conclusions

I propose a simple approach to exploit Twitter in building datasets of weak semantically similar texts. Results prove that curated paraphrases, such as in NLI datasets, are not necessary to train accurate models generating high-quality sentence embeddings since models trained on this datasets of weak pairs perform well on both established and novel benchmarks of informal texts.

The intrinsic relatedness of quotes with quoted texts and replies with the replied texts is particularly useful when building large datasets without human manual effort. Thus, I plan to expand the study to other languages spoken on Twitter. Two months of English data are more than enough to build large datasets, but the time window can be easily extended for rarer languages, as today, more than nine years of data are available to download. Finally, I hypothesize that this approach is adaptable to build high-quality embeddings for text classification tasks. I will extensively explore this on Twitter-related benchmarks.

In this section, I have trained a Language Model that accurately embeds texts in dense high-dimensional vectors. Since I train the model on texts from Twitter, it outperforms other approaches trained on formal data. This model can be used to perform other tasks related to Twitter. However, it can process only short texts. In the next section, I investigate how to compute accurate embeddings of users with a Hierarchical approach whose first stage is the model obtained in this section. I finally evaluate the obtained user embeddings visualizing communities, detecting outliers and quantifying controversies.

4.3.7 Ethical Considerations

We generate the training datasets and novel benchmarks starting from the general Twitter Stream collected by the Archive Team Twitter, as described in Section 4.3.2. They store data coming from the Twitter Stream and share it in compressed files each month without limits. This collection is useful since we can design and perform experiments on Twitter data that are completely reproducible. However, it does not honor users' post deletions, account suspensions made by Twitter, or users' changes from public to private. Using Twitter official API to generate a dataset is not a good option for reproducibility since parts of data could be missing due to Twitter Terms of Service. We believe that our usage of Twitter Stream Archive is not harmful since we do not collect any delicate information from tweets and users. We download textual data and connections between texts (quotes and replies), and we also remove screen names mentioned in the tweets during the cleaning step.

However, we agree that Twitter Stream Archive could help malicious and unethical behaviours through inappropriate usage of its data.

4.4 Hierarchical Transformers for User Semantic Similarity[1]

4.4.1 Introduction

Nowadays social media take significant parts of the lives of a large number of people, as the increasing number of active users and shared content testifies.[20] The magnitude of available data allows heterogeneous studies whose fields range from Social Network Analysis to Natural Language Processing [9,199,214,278].

The analysis of users' behaviours on social media platforms became an important branch of research since it allows customization of the overall personal experience [108]. Personalized experiences include connection recommendations [133, 143], usually implemented with features that suggest new friend/follow relationships, and targeted advertisement, whose goal is to pick the most relevant advertisements for each user. User profiling also helps to detect profile duplicates [270] and social threats [224], pointing at suspect behaviours that can be carefully investigated and quickly suppressed. Companies profile users for recruitment purposes to easily detect appropriate candidates for open job applications. Finally, community detection also benefits from user profiling. While it is a common and challenging task in graph analysis [109], where nodes represent users connected if they follow predefined rules (e.g., two users follow each other), neglecting the graph structure has its benefits. Graph-independent alternatives include clustering approaches using syntactic and semantic features of users [247] (Section 4.1).

Computing user similarity often requires heterogeneous information about users. Examples of selected features are the *textual-content* shared by users [207], the *social graphs* involving users (e.g., the follower/friend graph, the mention graph, or more advanced alternatives) [143], *shared links* and their source [248], *biographical* features (e.g., the profile picture, the biography that users can insert in the profile section and the selected geographical locations) [7], *demographic* features (e.g., gender, age and educational level) [155] and *numerical* features (e.g., total number of shared posts, the average number of shared posts per day, average number of likes and comments received and given and the number of followers/friends) [133].

However, the definition of similarity between users is not straightforward and becomes even more challenging when multiple kinds of features are involved. Two users could be very similar when inspecting demographic traits but different if we include the social graph or the shared content. The selection of the best combination of features is highly dependent on the final task selected. If our goal is to detect bots, then numerical features such as the frequency of shared tweets could be essential, but if the goal is to

[1] Authors: Marco Di Giovanni: first author, conceptualization, data collection, implementation, experiment design, writing; Marco Brambilla: critical feedbacks, editing.

[20] https://www.businessofapps.com/data/twitter-statistics/

recommend accounts to follow, then the social graph will be crucial.

In this work we investigate only the textual content shared by users and we leave for future works analyses on how to better combine it with other features.

The main advantage of working on textual data, independently of the underlying social graph that is commonly used in previous works, it that the former approach can detect similarities even between users that are far apart from each other or even belonging to different connected components of the graph. Moreover, the complete social graph is usually expensive to build due both to the magnitude of active users in the main social networks, and also because this information is usually slow and expensive to extract, affecting the speed of the whole analysis. Finally, we expect our approach to easily adapt to many text-based social networks, such as Twitter, Facebook and Reddit, while the graph structure is more dependent on the platform (e.g., the *follow* relationship from Twitter is much different from the *friend* relationship from Facebook, thus the corresponding graphs). However, we prioritize the quickness of the analysis to its holisticness, and we leave for future works detailed multi-platform analyses.

We select **Twitter** as the Social Networking site to investigate since it is worldwide used to communicate and stay informed, and it mainly relies on textual data. Our goal is to compute user embeddings that accurately reflect their semantic similarities. We train hierarchical models to map the textual content of tweets shared by users into high-dimensional dense vectors. We expect our map to transform similar users to vectors close to each other.

Classical semantic approaches rely on features such as Term Frequency–inverse Document Frequency (TF-iDF), often selected because of its simplicity, quickness and ease of implementation [247]. However, these techniques are usually too simple to obtain accurate results on challenging tasks.

Recently Natural Language Processing (NLP) field became highly popular due to the effectiveness of sophisticated approaches such as Transformer-based models [298] like BERT [91]. They exploit transfer learning techniques to obtain a state-of-the-art model by finetuning on small datasets a model pre-trained on enormous magnitudes of unsupervised textual data. It results in improvements on many text-related tasks such as text classification, text tagging and question answering. After the release of GPT-3 [51] text generation also became more reliable and coherent, and nowadays, the NLP community began to study robust Natural Language Understanding models [303].

In this work, we exploit the recent successes of NLP to generate high-dimensional semantic embeddings of users that encode similarities. We use Transformer-based models, previously pre-trained on Twitter texts, in a hierarchical approach [226]. We train our models with a triplet-like loss [153], where we minimize the distance between positive pairs of users and maximize the distance between negative ones. One of the main differences compared to previous works [207, 247, 248, 270], where a careful manual selection of small collections of users is likely to bias the results, is that we evaluate the models on a large set of users automatically obtained from Twitter. We can accomplish this due to a carefully designed evaluation process that does not require manually annotated labels. Our evaluation is fully reproducible since the datasets and the code will be publicly available, and the magnitude of the evaluation set makes the results statistically significant. We compare our models with baselines, and we extensively investigate the hyper-parameters to obtain the best configuration overall.

We formulate and answer the following research questions:

- **(RQ1)**: How can we evaluate the best model to compute semantic user similarity in a fully reproducible approach without influencing the results with biased selections of small sets of users?
- **(RQ2)**: How can we apply to tens of posts a Transformer-based model, widely known to be effective on single texts or pairs of texts with limited length?
- **(RQ3)**: Do the obtained embeddings reflect our idea of similarity? Can we use them for further tasks?

Our contribution is four-fold:

1. We collect a large dataset of Twitter users, we design an automatic labelling approach, and we share the code and the parameters to reproduce it;
2. We train and release a Hierarchical Language Model to compute accurate user similarity;
3. We extensively investigate hyper-parameters to obtain the best configuration of the model;
4. We test whether the obtained embeddings are accurate when applied to other tasks.

4.4.2 Related Work

In this section we summarize the principal related works about Twitter user similarity and state-of-the-art language models. To the best of our knowledge, no previous studies applied Transformer-based Language Models to user profiling and similarity.

User Profiling and Similarity

User profiling is a research field whose purpose is to profile users of a platform to personalize their experience. Many researches investigated the best techniques to model behaviours of users, usually highly dependent on the platform. Here we report and describe some of the main successful approaches to profile Twitter users and to compute similarities. We distinguish between approaches relying on multiple heterogeneous features (Comprehensive) and approaches based solely on content.

Comprehensive Approaches Comprehensive approaches require multiple features to compute similarities. These approaches are often slower and their performance is not guaranteed to be higher since similarity is usually poorly defined in this context.

Twitter research team firstly proposed **Who-To-Follow** [143] approach, whose goal is to recommend potential users to follow. It is based on common connections and shared interests. The core of the architecture is Cassovary, a graph-processing engine that implements the main graph recommendation algorithms, applied to the Twitter "follow" graph.

They also proposed a **similar-to** framework [133] where similar users are detected by comparing four signals: cosine follow score, number of suggestions' followers, page rank score, and historic follow-through rate. They train a Logistic Regression model to learn whether two users are similar. Their approach includes also a candidate generation step and it is highly scalable. They evaluate the approach using both human annotators and follow-through rate (percentage of the follows among all the impressions of the suggestion).

An alternative approach [299] is designed to include Affinity Propagation to obtain communities. Similar users are defined with several metrics based on shared content, following relationships and interactions, follower, friend, hashtag, reply and mention similarities. The clusters are analyzed with LDA to obtain topics discussed by the communities. The authors validate this model with a single experiment involving about $3k$ users.

TSIM: a system for discovering similar users on Twitter [7], is a framework to quickly detect users similar to a specific user by computing seven different signals: following and follower, mention, retweet, favourite, hashtag, interest (subjects and subsubjects extracted from an online English dictionary) and profile (gender, language and location) similarity. They combine these similarities with manually picked configurations of weights. Their algorithm exploits MapReduce [86] model to process large number of candidate users. They evaluate the system using human judges, even if they state that they "are difficult to rely on when measuring the accuracy of the system" and comparing tens of users to outputs of "Who To Follow" Twitter service, whose outputs are randomly selected thus not completely reproducible.

We do not compare our approach to comprehensive approaches since they require data that are not available in our collection.

Content-based Approaches Content-based approaches are preferred since usually content of tweets is quickly available and the definition of textual similarity has been recently largely investigated. Our proposed approach is content-based and we define similar users as users that share semantically similar tweets.

Twitter-based User Modeling Service (**TUMS**) [283] is a service that generates semantic user profiles by exploiting tweets. Given a user ID, it collects and processes its tweets to produce entity-based, topic-based or hashtag-based profiles. It also semantically enriches tweets by extracting entities from related external Web sources such as URLs. Finally it allows to see which topics or entities a user was interested in at a specific point in time. Unfortunately, the code is not publicly shared and the official API is no longer available, thus we were not able to compare this approach is our experiments.

A different approach to compute content-based similarities of Twitter users [207] is performed by building a network representing semantic relationships between words occurring in the same tweet. The graph is exploited to compute network centrality measures and finally user similarity is computed with cosine similarity. The evaluation involves 17 famous Italian users and the results are validated qualitatively. The authors did not share the code being in a prototypical phase, and to the best of our knowledge, a stable implementation was never published. We were not able to implement their approach due to the strong dependence on [208], a not available text enriching procedure designed by the same authors.

Finally, different from Section 4.1 where we applied TF-iDF vectorizations on different kinds of words, such as Nouns, Verbs, Proper Nouns and DBPedia Instances and Topics, here we exploit recent deep Language Models. Detailed results are reported below.

One of the most important differences between our study and previous works is the magnitude of the number of users involved in the experiments, without the need of human annotators, thanks to the carefully designed evaluation strategy.

Language Models

Language models became enormously popular after the surprisingly accurate performances of **BERT** [91], a deep Transformer-based [298] model pretrained in an unsupervised way on large corpora of text using two self-supervised techniques: Masked Language Models (MLM) task and Next Sentence Prediction (NSP) task. The model is designed for transfer learning: it has to be finetuned for a few epochs for specific tasks, inserting an additional fully-connected layer on the top, without any substantial task-specific architecture modifications. The BASE version of BERT contains 12 layers with 768 hidden dimension and 12 heads per layer, for a total of $110M$ parameters (see Section 2.4.4 for further details).

RoBERTa [192] is an improved pre-training of BERT-base architecture, including dynamic masking, Next Sentence Prediction formats, larger batch size and a BPE vocabulary of $50K$ subwords units. It was trained on the same data as BERT but it results in a much robust model (see Section 2.5.1 for further details).

BERTweet base [218] is a BERT-base model pre-trained using the same approach as RoBERTa on $850M$ English Tweets, outperforming previous SOTA on Tweet NLP tasks.

Sentence BERT [252] are BERT-base models trained with siamese or triplet approaches on Natural Language Inference (NLI) and Semantic Textual Similarity (STS) data. We select a suggested base model from the full list of trained models: stsb-roberta-base-v2 (S-RoBERTa).

Twitter4SSE exploits Twitter's intrinsic powerful signals of relatedness (quotes and replies) to generate semantically similar embeddings training a Transformer model with a triplet approach [95] (see Section 4.3 for further details).

Recently, many alternatives have been proposed to process **long texts** with variations of BERT. Some works focus on reducing the computational complexity of attention, so that instead of scaling with the square of the number of tokes, it scales linearly [22,322].

An alternative is performed by using a hierarchical approach to process long texts. **Hierarchical Transformers** [226] are hierarchical models used to combine chunks of long texts into a single embedding. After using BERT to embed single chunks, they train two models on the top of it, namely RoBERT (Recurrence over BERT) using LSTM layers, and ToBERT (Transformer over BERT) using Transformer layers (see Section 2.5.3 for further details).

4.4.3 Data

In this section we briefly describe how we collected and cleaned raw data, and the selection of users to include in the training and evaluation datasets. We focus on the first research question **(RQ1)** while designing an approach to build large datasets in a fully reproducible way.

Data Collection

We select Twitter as the social media platform on which we perform our analyses, but the whole approach and experiments can be easily transferred to other platforms with few small changes. Twitter is a widely used microblogging platform, where people can easily subscribe and post short texts (max 280 characters) with pictures and urls attached. Users can *follow* other users to easily see what they post. Users can *post* original tweets or *reply* to other tweets. Users can also share a tweet posted by other users with or without an original comment (respectively called *quote* or *retweet*).

Retweets typically indicate endorsement [24, 42]: when a user retweets a tweet posted by a different user, it agrees with the content shared by the retweeted user. We use this statement to build our groundtruth for similar users: two users are similar if at least one of them retweeted at least once a tweet shared by the other. We remove clear exceptions of extreme behaviours, such as users retweeted by too many users, in the cleaning step. This hypothesis is not always true when quotes are involved, since the original comment attached to the tweet could include criticisms and objections.

We build our dataset from **Archive Team Twitter**[21]. We do not download tweets from Twitter official API since it does not guarantee reproducible results: the same request made at different times and by different accounts could result in different collected data. Twitter Stream Achieve shares data from the official Twitter Stream, but does not honor users' post deletions, account suspensions made by Twitter, or users' changes from public to private, i.e., the data are collected in files that are not updated.

We remark that we do not include delicate information in our analysis. The full approach involves only the textual content shared by users, cleaned as described later by removing the screen names of users. We use the ids of users only to group tweets shared by the same user. We do not share the obtained datasets but we will share the full code to reproduce them from the Twitter Stream Archive files. However, we agree that Twitter Stream Archive could help malicious and unethical behaviours through inappropriate usage of its data.

We select English tweets, filtered accordingly to the "lang" field provided by Twitter, posted in November and December 2020. They amount to about $27G$ of compressed data.

We collect texts of tweets, including replies and quotes but excluding retweets, and ids of users that posted them, and, in parallel, we collected pairs of ids of users if one of them retweeted the other. This second collection is performed to obtain a groundtruth of similar users, since we assumed that users that retweet each other are similar.

We collect a total amount of $38M$ tweets and $95M$ pairs of users from retweets.

Text Cleaning

Before performing any analysis, we strongly clean the obtained texts, since tweets are known to be extremely noisy. We remove mentions (appearing as the symbol @ followed by a screen name) and urls, frequently attached to tweets. We standardize spaces replacing tabs, newlines and multiple consecutive spaces with a single whitespace and we lowercase the full texts. Finally, we remove texts shorter than 20 characters, since they are too short and do to not contain enough information to be processed (single word tweets are common especially in replies, often followed by urls and/or mentions).

After the cleaning procedure about $29M$ texts tweeted by $10M$ unique users remain.

User selection

Since we collected data tweeted during a 2-months window, we remove users that posted too many tweets, since they may be bots. We set the maximum number of tweets to 60, being about 1 per day. We also remove users with less than 5 tweets

[21] https://archive.org/details/twitterstream

collected, as we do not have enough information to perform the following analysis on them. We obtain $1.4M$ different users that we define *Good Users*.

We also clean the connections between users, removing from pairs of ids of users retweeting each other the auto-retweets (when a users retweets one of its own tweets), duplicate pairs (when a user retweets more than once another user or when two users both retweet each other) and pairs where at least one of the users in not what we have previously defined as Good User.

Finally we remove users with more than 50 connections, since they represents accounts that retweets or are retweeted too much to be considered similar to all of the other users. We finally obtain about $1.9M$ connections between $950k$ unique users.

Training and Evaluation Datasets

We firstly generate our evaluation dataset by randomly selecting 5k users with at least 5 connections, each one paired with 5 randomly selected users from the connected ones. Our final evaluation benchmark consists of comparing a user with 30 other candidate users, 5 of them considered similar to it since they share at least one retweet connection, and 25 of them considered not similar, randomly selected among the other users.

Even if we agree that evaluating a user similarity model using groundtruth based solely on a single retweets connection is a strong assumption, we remark that we only assume that two users connected by a retweet relationship are *more* similar than two randomly selected users.

Finally, we remove connections involving the previously selected users and we create the training set with the remaining pairs of connected users. From each user we collect the first n tweets posted. This number is analyzed in detail in the Evaluation Section and defines the final size of the training dataset.

4.4.4 Methodology

In this section we describe our hierarchical approaches and some baselines. We also include brief technical details to support complete reproducibility of the results.

Our approach is inspired by [226], where the authors build a hierarchical model to process long documents. They exploit a hierarchical approach to process a long single document that cannot fit into a standard BERT architecture due to length limitations. Instead, our data are multiple short documents (i.e., texts of tweets) from the same user, which naturally fits the hierarchical structure.

Thus we answer the second research question **(RQ2)** by building a hierarchical Transformed-based approach.

Hierarchical Approaches

Our hierarchical approaches are composed by two stages, a tweet embedding stage and a user embedding stage. The complete pipeline is schematized in Figure 4.6.

Stage-1 We obtain embedding of tweets using one of the following four Transformer-based models that share the same architecture but are pretrained with different approaches and datasets. We test them both freezing and unfreezing their weights during the training step.

- **RoBERTa**[22]: a baseline model pretrained on Masked Language Model task with carefully selected techniques that improve the performance of its predecessor: BERT;
- **BERTweet**[23]: it performs the same pretraining as RoBERTa but on a big dataset of texts from Twitter;
- **Sentence BERT**[24]: a model initialized from RoBERTa and trained on Semantic Textual Similarity tasks with a Siamese architecture. The obtained model reaches state-of-the-art-results in Semantic Sentence Embedding tasks, evaluated with cosine similarity;
- **Twitter4SSE**[25]: a model initialized from BERTweet and trained with an approach similar to Sentence BERT on Twitter data. The obtained model reaches state-of-the-art results in Tweet Semantic Similarity tasks.

BERTweet and Twitter4SSE models, being pretrained on texts from Twitter, are able to succesfully deal with the intrinsic noise of data from social media, thus no further special cleaning is required (such as dealing with hashtags, abbreviations, and typos).

Stage-2 We test three techniques to process twitter embeddings to generate accurate user embeddings:

- **MEAN**: the simplest approach to merge tweet embeddings into a fixed size vector representing user embeddings is to compute their MEAN. This approach can be performed without limits of the number of tweets n per user when the weights of the Stage-1 model are frozen (no training is performed when we select this variant). However we test this approach also unfreezing the weights of the Stage-1 model, thus we limit the number of tweets per user, also for a fair comparison with other variants;

[22] https://huggingface.co/roberta-base
[23] https://huggingface.co/vinai/bertweet-base
[24] https://huggingface.co/sentence-transformers/stsb-roberta-base-v2
[25] https://huggingface.co/digio/Twitter4SSE

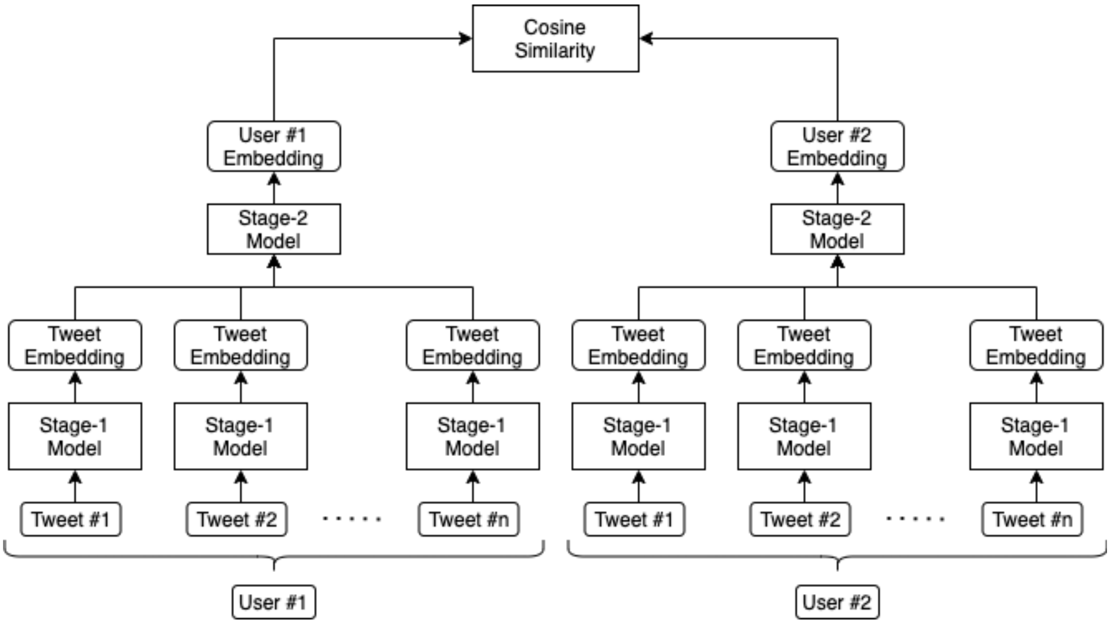


Figure 4.6: *Schema of the Hierarchical approaches.*

- **Recurrence over BERT (RoBERT)**: the embeddings of tweets are used as input of a Recurrent Model. We select a 2-layer LSTM model with hidden size of 768.[26] We use the last output as the user embedding. We test this approach both freezing and unfreezing the weights of the Stage-1 model;

- **Transformer over BERT (ToBERT)**: the embeddings of tweets are used as input of a Transformer Model with 2 encoding layers (EL) and 2 decoding layers (DL), 16 heads and 0.1 dropout. We also experimented with a model with 1 encoding and 1 decoding layer and without dropout (more details are reported below). Transformers output one embedding for each input, so we select the MEAN of all output embeddings as the user embedding. We test this approach both freezing and unfreezing the weights of the Stage-1 model.

Non-Hierarchical Approaches

We compare our approaches with two simple non-hierarchical alternatives:

- **TF-iDF**: Term Frequency–inverse Document Frequency is a classical vectorizer of documents belonging to a corpus. It can be applied to documents of any length. We consider as single document the concatenation of the tweets of a single user. We use the TfidfVectorizer implemented in scikit-learn [228], testing with or without bigrams, with or without English stopwords and different values of minimum document frequency. We report results of the best set of hyperparameters, but we

[26] Preliminary experiments with different number of layers show evident advantages with respect to a single layer architecture, but not clear improvements when using deeper architectures.

remark that the choice of them does not influence significantly the overall performance. We remark that this approach is not trainable by definition;

- **Naive Transformers**: similar to TF-iDF, we consider a single document the concatenation of the tweets of a single user. We tokenize the documents and use them as inputs to a Transformer model, which truncates them at 128 tokens. We test four Transformer models previously described.

We select **Multiple Negative Loss (MNLoss)** [151] as our loss funcion for every trainable model: given a batch of positive pairs of users $(a_1, p_1), ..., (a_n, p_n)$, we assume that (a_i, p_j) is a negative pair for $i \neq j$ (i.e., we assume that a user did not retweet posts from any of the other $n - 1$ users). This assumption is valid for small batches due to the big total number of users and the approach selected to collect data. The probability of having a user connected to a different users in the same batch is negligible for batch sizes that fits in memory. We minimize the negative log-likelihood for softmax normalized scores. We expect the performance to increase with increasing batch sizes, thus we set $n = 60$ when the weights of the Stage-1 model are frozen and $n = 10$ when we also finetune those parameters, being the highest values that fits in memory.

We train the models for one epoch (more details are reported below). We use AdamW optimizer, learning rate 2×10^{-5}, linear scheduler with 10% warmup steps on a single NVIDIA Tesla P100.

4.4.5 Evaluation

In this section we report and discuss results of our models compared to variants and baselines to find the best approach overall. When not stated differently, we use 20 tweets per user, thus $124k$ pairs of users in the training set. More details about dataset sizes are reported in Figure 4.7.

Metrics

We evaluate the models by comparing three metrics, commonly used for similar tasks. These metrics evaluates different aspects of the rankings and we generally obtain compatible scores.

- **Mean Average Precision (MAP)** between the binary labels (connected or not connected by retweets) and the similarities. It summarizes a precision-recall curve. In this setting it ranges from 0.17 when the 5 connected candidates receive similarity score of 0 and the 25 not connected candidates receive similarity score of 1, to 1, when the similarities are the opposite;

- **Mean Reciprocal Rank (MRR) @k** is a ranking quality measure defined as the reciprocal of the rank of the first relevant element, if not greater than k. We set

Model	MAP	MRR@10	nDCG
RoBERTa	81.1	94.2	90.8
BERTweet	83.7	95.3	92.2
S-RoBERTa	81.3	94.4	91.0
Twitter4SSE	**84.2**	**95.6**	**92.4**

Table 4.15: *Comparison of Stage-1 models*

$k = 10$. MRR@10 ranges from 0, if none of the 5 connected users are ranked in one of the first 10 positions, to 1 if the most similar user is connected;

- **normalized Discounted Cumulative Gain (nDCG)** [161] is a ranking-quality metric obtained normalizing Discounted Cumulative Gain (DCG). In this setting the scores range from 0.35 to 1, the higher the better. Thus, 1 represents a perfect ranking: the first ranked document is the most relevant one, the second ranked document is the second most relevant one, and so on.

Stage-1 Model Comparison

Firstly we investigate the best initialization model. For each experiment we keep the same hyper-parameters and the same Stage-2 model is trained on the top of it: ToBERT with 2 encoding layers (EL) and 2 decoding layers (DL), 0.1 dropout and MEAN pooling. We test RoBERTa, BERTweet, S-RoBERTa and Twitter4SSE. Table 4.15 shows that Twitter4SSE is the best initialization. As expected, this model, trained to generate accurate tweet embeddings, outperforms both the model trained on Tweets using only MLM (BERTweet) and the model trained to generate accurate sentence embeddings on formal data (S-RoBERTa).

MEAN Stage-2 Models Comparison

We test the MEAN Stage-2 approach on the four Stage-1 models with and without freezing their weights. Table 4.16 shows that unfreezing the weights leads to better results, even if the batch size has to be reduced to 10 and the number of tokens per tweet is reduced to 32 to fit in memory. We confirm that the best Stage-1 model is Twitter4SSE for these configurations too.

ToBERT Hyperparameter Comparison

We investigate the best hyperparameter configuration of the Stage-2 Transformer model (ToBERT). We investigate with 1 and 2 encoding and decoding layers (EL-DL), with and without dropout. We fix Twitter4SSE as initial model. Table 4.17 shows that 2 EL and 2 DL without dropout is the best overall configuration.

Stage-1	Frozen	MAP	MRR@10	nDCG
RoBERTa	yes	33.3	64.0	60.1
S-RoBERTa	yes	33.2	63.8	60.1
BERTweet	yes	33.3	63.7	60.7
Twitter4SSE	yes	33.3	64.0	60.1
RoBERTa	no	80.8	94.3	90.7
S-RoBERTa	no	81.2	94.7	91.0
BERTweet	no	83.1	95.3	91.9
Twitter4SSE	no	**83.6**	**95.8**	**92.3**

Table 4.16: *Comparison of MEAN Stage-2 models*

Model	EL-DL	D	MAP	MRR@10	nDCG
ToBERT	1-1	0.1	84.3	95.8	92.6
ToBERT	2-2	0	**84.5**	**96.0**	**92.7**
ToBERT	2-2	0.1	84.2	95.6	92.4

Table 4.17: *Comparison of ToBERT Stage-2 models*

Full Comparison

We compare the performance of the models with a Random baseline[27] and with the two best approaches from [247][28].

Naive approaches underperform Hierarchical approaches confirming an advantage to encode single tweets independently. The hierarchical approach with a Stage-1 Twitter4SSE model and a Stage-2 Transformer model outperforms the other alternatives. We notice a gap of performance with respect to the same model with a Stage-2 LSTM model, empirically proving the goodness of transformer layers with respect to recurrent layers in this setting, while confirming that the sequential nature of tweets is not critical. TF-iDF best approach (including bigrams, excluding english stopwords and words that appeared less than 5 times in the whole dataset) is comparable to Naive Transformers, whose alternative initialized from Twitter4SSE is the only model that clearly outperforms the not-trained baseline. However, its performances are far from the Hierarchical alternatives. PROPN [247] does not improve classical TF-iDF approaches while LDA [247] reaches performances marginally higher than a random baseline in our setting. Finally, as expected, when unfreezing the weights of Stage-1 models, the performance increases even if we had to reduce the batch size (models denoted with _fr are the frozen variants).

Selection of Number of Tweets per User

Now we inspect the number of tweets to process per user n. To perform a fair comparison, we build the test set as described in the Data Section with users that have tweeted at

[27] the similarity between two users is a random number uniformly distributed between 0 and 1

[28] We select the best syntactic and semantic approach: PROPN stands for Tf-iDF only on proper nouns, and LDA is firstly analyzed by checking the highest coherence score, selecting 14 topics.

Model	MAP	MRR@10	nDCG
Random	25.3	36.0	51.8
TF-iDF	59.6	79.3	77.9
PROPN [247]	59.5	79.3	77.8
LDA [247]	30.4	45.7	57.1
Naive RoBERTa	55.5	85.1	76.9
Naive S-RoBERTa	42.4	73.6	60.7
Naive BERTweet	53.0	82.3	75.1
Naive Twitter4SSE	70.1	90.1	85.1
Twitter4SSE MEAN	83.6	95.8	92.3
RoBERT_fr	79.8	94.0	90.2
RoBERT	83.0	95.1	91.8
ToBERT_fr	82.3	94.9	91.5
ToBERT	**84.5**	**96.0**	**92.7**

Table 4.18: *Full Comparison of Models and Baselines.*

least 30 tweets, and we keep it fixed for every experiment. Firstly we check if a greater number of tweets per user implies a better accuracy by selecting all the remaining users with at least 30 tweets as training set. We train models by changing n and we report the results in Figure 4.7 (right, red crosses). As expected, a greater number of tweets per users results in a better model, when the number of pairs of training users is fixed.

However, in the Figure 4.7 (left) we show how a greater n implies a lower number of users since we have a limited collection of tweets. Thus we investigate what is the best trade-off between the number of users and the number of tweets per user. Figure 4.7 (right, green squares) shows the performance of models trained changing the number of tweets per user, including every user available. A peak around 20 tweets suggests our best trade-off. However, we remark that this number is highly dependent on our collection, since the number of downloaded tweets is high but finite (2 complete months). Expanding the collection time window will result in a different trade-off, since we could be able to collect a greater number of users. We remark that Figure 4.7 (right, red crosses) shows that the performances plateau at about $n = 25$, thus expanding the collection window will influence the results but not the best values of number of tweets per user to process.

Selection of Number of Training Epochs

One could argue that a lower number of users could be compensated increasing the number of epochs. We investigate how this hyper-parameter influences the results in Figure 4.8.

The left plot shows how simply increasing the number of training epochs scarcely improves the performance of the model, while the computational time linearly increase. Moreover, the performance immediately reaches a plateau and the model risks overfitting due to the lack of an evaluation dataset.

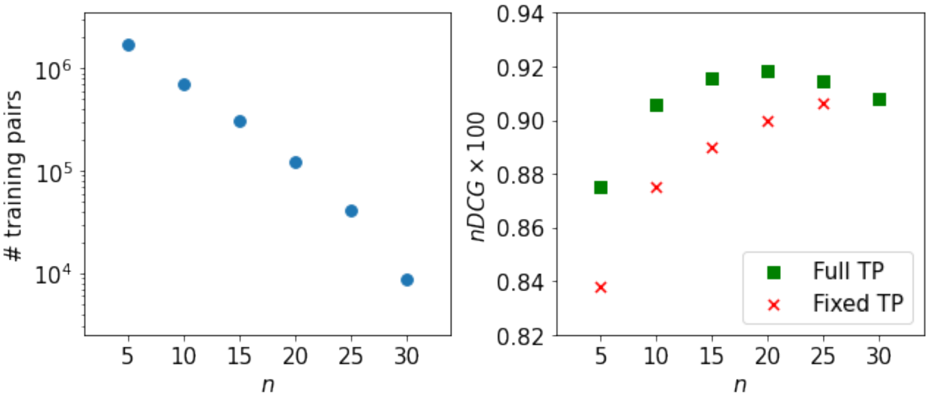


Figure 4.7: *Left: Distribution of the total number of pairs of users in our collection with respect to the minimum number of tweets posted n. Right: $nDCG \times 100$ of models (red crosses) when the training set is fixed and (green squares) when we include the full set of pairs of users.*

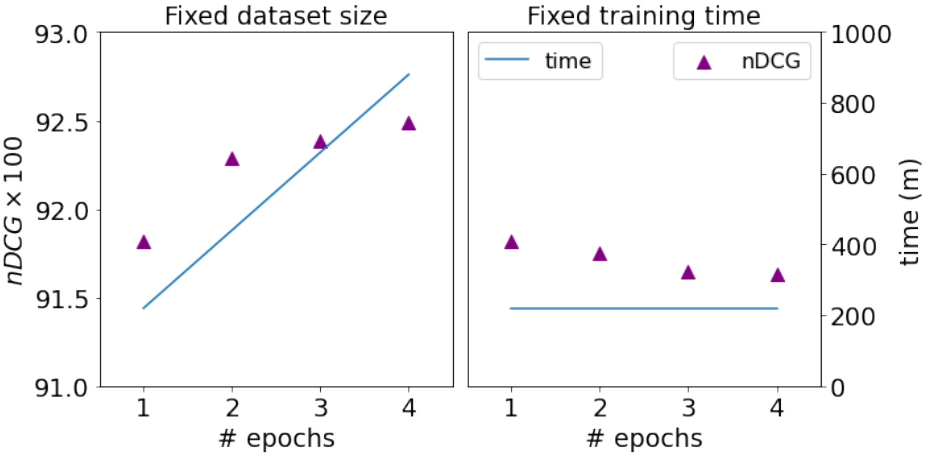


Figure 4.8: *$nDCG \times 100$ score (purple triangles) and training time in minutes (blue lines) of models trained for different epochs. (left): Fixed training dataset; (right): Reduced training dataset to keep the training time constant.*

The right plot shows the performance when the dataset is reduced to keep the training time constant, e.g., when the number of training epochs is set to 2, we reduce the training set by a half, when the number of training epochs is set to 3, we remove two thirds of the training data and so on. As expected, increasing the number of epochs does not compensate the removed data. This implies that, if we would aim for a better model, increasing the time window selection should be preferred to increasing the number of training epochs.

4.4.6 Further Analyses

In the previous section we verified that our fine-tuned hierarchical architecture generates user embeddings that highly correlate with the similarity between users defined from the "retweet" relationship. In this section we answer the third research question

(RQ3) by checking whether the obtained embeddings and the similarity scores actually reflect our idea similarity between users, and how to use these embeddings for other tasks, such as community visualization, outlier detection and polarization direction detection. The results of this section prove that our main assumption is valid.

In this section we exploit **Twitter lists** to generate sets of similar users. Twitter lists are collections of users' handles, called members, and they can be followed by other users to get activities related to the members of the list in their feeds. Everyone can create a Twitter list and set it as public or private. Usually lists include carefully selected members that tweets about the same topics, thus the users that follow the lists obtain a coherent feed, easy to browse. We select a set of public lists reported in Table 4.19, where M is the number of members of the list and F is the number of followers, that usually reflects the goodness of the list. We manually selected those lists by browsing the ones created by the Twitter account @*verified*, an official Twitter account that manage the blue badge of verified accounts, by @*tweetcongress*, a Twitter account that shares tweets about the US Congress and by other reliable accounts grouping official accounts of sport teams and clubs. We check that no user belongs to more than one of our selected lists. For each user in the list, we collect the last 20 tweets shared before 28/09/2021, including retweets.

The whole analysis is performed using a hierarchical model with a frozen Stage-1 Twitter4SSE model and a Stage-2 ToBERT model with 2 layers, 0.1 dropout rate, MEAN pooling, trained using 20 tweets for each user for one epoch.

Name	**Owner**	M	F
NBA teams	@chicagobulls	31	360
NFL (Teams)	@Sportsguy786	32	443
Clubs	@MLB	43	3759
NHL Team Accounts	@NHL	37	9190
Chess Grandmasters	@chesscom	36	18
technology	@verified	20	1437
foodies	@verified	11	500
charity-ngo	@verified	23	631
Democrats	@tweetcongress	51	262
Republican	@tweetcongress	108	1118

Table 4.19: *Selected lists of users. The number of members M and number of followers F are reported as they were on 28/09/2021.*

Community visualization

We visualize reduced embeddings performing PCA with two principal components. Figure 4.9 shows an example of PCA applied to embeddings of the five communities of Table 4.19 about sports (first five lines: NBA collects Basketball Clubs, NFL Football Clubs, MBL Baseball Clubs, NHL Hockey Clubs, and Chess players). We observe that members of the same lists are clustered, even if the topics of the selected communities are similar, since they are all related to sports. As expected, the community of Chess players is more separated to members of other lists, being chess a much different sport.

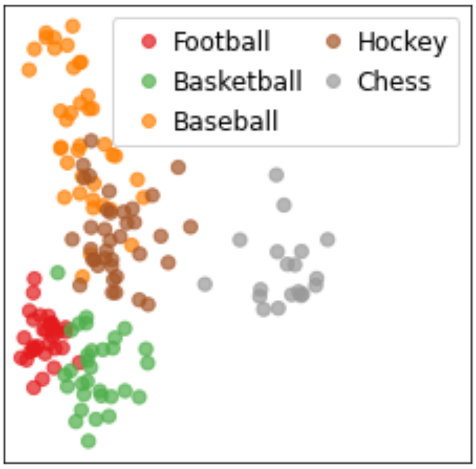


Figure 4.9: *First and Second Principal Components of members of five sport-related lists.*

Outlier Detection

Members of a list do not always share on Twitter content about the same topic. This happens because not everybody uses Twitter to tweet about the topic that they are known for. We can use our model and the obtained user embeddings to detect outliers, i.e., users that are not similar to users in the community for the content shared, even if they are members of the same list. We tested this approach by applying Local Outlier Factor (LOF) [49] algorithm on three lists of users and we manually inspected the results. When applied to embeddings of *technology* list, it outputs one outlier (@majornelson). After manually inspecting the 20 processed tweets, we discover that this account mainly tweets about videogames, while the rest of the members share technology news in general. When applied to the *foodies* list, LOF outputs only @BryanVoltaggio as outlier, a chef that mainly retweets about topics different from food. When applied to *charity-ngo* list, we obtain the official account of Charlize Theron (@CharlizeAfrica), an actress that founded Charlize Theron Africa Outreach Project (CTAOP). We clearly notice that even if some of the tweets from the personal account are actually related to CTAOP, her feed is obviously much different from other NGO members of the list.

Polarization Direction Detection

Given two communities that are polarized by definition, we use the embeddings to define a one-dimensional subspace of the 768-dimensional space of embeddings, that represents the polarization direction. In this example, we pick the list of Democrats and the list of Republicans as the two polarized communities. We perform Linear Discriminant Analysis (LDA) [149] to obtain the one dimensional projection of the user embeddings. Figure 4.10 shows how the users of the two political parties are projected in the new subspace. However, we can potentially project every user and obtain their expected inclination. As expected, the distributions are clearly separated and generate a linear subspace quantifying the polarization.

This approach can be extended using every pair of communities that defines a polarizing topic, similar to the American structure of politics, but not when the sides are more than two. We can also use this approach when we have only two users, instead of two lists of users, to generate the polarization direction, for example selecting only the most representative accounts for each side of a debate.

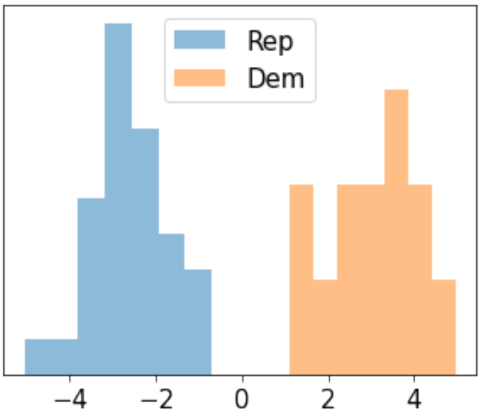


Figure 4.10: *Histogram of one-dimensional LDA projections of members of Republicans or Democrats lists.*

4.4.7 Conclusion

In this work, I studied the best approach to embed users so that the obtained vectors reflect our idea of similarity. I restrict the analysis on Twitter and, in particular, on the textual content of tweets, leaving analyses of other social networks and features (graph, biographical, demographic, temporal, ect.) as future works. I designed an scalable approach to obtain large pairs of similar users exploiting the "retweet" feature of Twitter without human annotators. I verified that a Hierarchical Transformer model outperforms classical and straightforward approaches, and I performed ablation to check the best initialization model and hyper-parameters. I evaluated my work on a large validation set to obtain statistically significant results. Finally, I applied the obtained embeddings to other tasks, e.g., visualization of communities and outlier detection, confirming that they reflect our concept of similarity. I trained and released the models so that they can be downloaded for further research. I share our code so to promote complete reproducibility of the obtained results.

Future works include the evaluation of the influence of the selected time window to the user embeddings, since topics discussed on Twitter are highly related to contemporary events. I also plan to investigate how the embeddings differ when I analyze tweets shared by the same user on different dates. Finally, since the whole approach, including the dataset generation, is language independent, I plan to evaluate it on other languages spoken on Twitter.

In this chapter, I analyzed the best approaches to process and analyze Twitter users. Starting from simple TF-iDF alternatives described in the first section, in the last section I trained an accurate Hierarchical Transformer model that produces dense user embeddings that reflect our idea of similarity. The following chapters describe applications of NLP techniques to contemporary events. The selected studies are mainly at the tweet level, and they do not exploit Hierarchical Transformer models proposed here to embed users. However, I plan to apply them to further analyses on the same datasets or novel contemporary events.

CHAPTER 5

Applications: Politics

In this Chapter, I report two studies about investigations of NLP techniques applied to political discussions on Social Media.

In the first work, I describe a simple classical approach to classify the political inclination of politicians performed after the 2018 Italian election where four main parties were present in the parliament. The obtained embeddings give insights into how deputies use Social Media. However, the approach is hard to transfer to citizens.

In the second work, I apply modern Language Models to classify tweets about the 2020 Italian Referendum. It results in insights into the discrepancy between social media activity and the outcome of the referendum. This work focuses on single tweets whose ground truth comes from hashtag-based semi-automatic labelling.

I decided to focus on Italian politics both because the topic is nearer to my interests and because I believe that the obtained and shared datasets could help the NLP community due to the low number of non-English datasets available today.

If the social activity of Italian citizens on Twitter will not decrease in the future, with recent advanced techniques, it will be possible to obtain even more accurate predictions and insights from the publicly shared content, threatening the privacy of people. The obtained results could be dangerous and help malicious behaviours.

5.1 Content-based Classification of Political Inclinations of Twitter Users[1]

5.1.1 Introduction

Classification of political inclination of people in social networks is a topic that collects increasing attention after the 2016 USA election and with the incoming ones. Similar analyses have been performed in other countries where social networks are widely used for political discussions and propaganda during electoral periods.

Estimating political inclination of people by looking at what they share on social media is a useful tool to predict election results, since it can be compared to statistics obtained with classical methods such as surveys at a lower cost.

Many kinds of information can be extracted from social networks, ranging from user connections (friend and follow relationships) to temporary interactions (likes, comments, reposts) to biographical information (geographical location, profession, education) and content posted (texts, images, videos, links).

Social networks collect wide amounts of data that are constantly updated in real time, so that temporal analysis can be performed. The detection of changes in behavior when critical events approach is useful to understand and forecast the reaction of people.

We focus on textual data since our goal this Chapter is to understand if written content can be used for classification tasks and for similarity scores computation, as opposed to classical methods that analyze the social networks as a graph.

In Italy, the most used Social Networks are Facebook, Instagram and YouTube. However, since our analysis focus on syntactic proprieties, we chose Twitter, mainly used for sharing texts. Moreover, Twitter public API allows to easily extract data, with limits only in terms of frequency. Twitter is widely used to discuss political issues in Italy, where politicians and supporters perform low-cost propaganda and constantly share their opinion.

Our main hypothesis is that texts in tweets contain enough information to classify users by their political inclination. To test this hypothesis, a not-easily collectible ground truth is needed, since individuals rarely share their political inclination on social networks (thus, the secretiveness of the vote). Hence, to first evaluate different

[1] Authors: Marco Di Giovanni, first author, conceptualization, implementation, experiment design, writing; Marco Brambilla, conceptualization, editing; Stefano Ceri, conceptualization, editing; Florian Daniel, conceptualization, editing; Giorgia Ramponi, conceptualization, experiment design, editing. [96] ©2018 IEEE. Reprinted with permission, from Marco Di Giovanni, Marco Brambilla, Stefano Ceri, Florian Daniel, Giorgia Ramponi, Content-based Classification of Political Inclinations of Twitter Users, 2018 IEEE International Conference on Big Data (Big Data), December 2018. In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Politecnico di Milano's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

prediction methods, we chose to perform the analysis on politicians, whose political inclination is obviously public.

An important aspect to consider is that since we are not using the social network structure, we are not confined to classify people connected to each other. Any user that has at least a moderate activity on Twitter can be classified, ignoring its social connections. Of course, incorporating the social network structure could be done to improve the prediction accuracy, but it can also limit the prediction power since the account must be in some way connected to others, requisite not needed in the context of this work.

Unlike classical tools such as surveys, algorithms based on social network analysis are faster and cheaper and can be used on a larger scale with a relatively small effort. Large quantities of data can be collected daily obtaining a wider and more heterogeneous set of people analyzed. However we have to deal with a social network bias, since the Twitter community is not always homogeneously distributed with respect to voters. It should be taken into account that people belonging to different parties can represent different classes in the society, thus can be more or less inclined to use social networks as a political instrument. The analysis of the gap between the real distribution of voters and the one predicted by an algorithm trained on social media data can give interesting insight on the voters of different parties.

For research purposes, we chose to split the Italian political situation of 2018 in four groups: “MOVIMENTO 5 STELLE", “LEGA - SALVINI PREMIER", “PARTITO DEMOCRATICO" and “FORZA ITALIA - BERLUSCONI PRESIDENTE". Other parties have been discarded since the fraction of population that voted them was small enough not to be relevant for our purposes.

Italian political situation in 2018

After the elections of March 2018, the Italian government was composed of a coalition between “MOVIMENTO 5 STELLE" and “LEGA - SALVINI PREMIER", with respectively 32,7% and 17,4% voters at the elections in March 2018. The other two main important parties are “PARTITO DEMOCRATICO" and “FORZA ITALIA - BERLUSCONI PRESIDENTE", with respectively 18,7% and 14,0% voters at the election. Thus, the four selected parties represent the 82,8% of the total voters.

Tweets used for this work are collected in August 2018.

The chamber of deputies is composed by 630 members, divided between parties with relation to the percentage of votes obtained. In Table 5.1, the actual numbers are reported.

Table 5.1: *Italian Deputies Chamber: distribution of deputies among the four major parties after the election on March 4, 2018.*

Party	Deputies	Fraction
MOVIMENTO 5 STELLE	221	35.1%
LEGA - SALVINI PREMIER	125	19.8%
PARTITO DEMOCRATICO	111	17.6%
FORZA ITALIA - BERLUSCONI PRES.	105	16.7%
Other minor parties	68	10.8%

5.1.2 Related work

In recent years, many researchers analyzed social network accounts to obtain information about political inclination of users. Often, analyses are made before and around election days, to obtain insights and predictions of the results.

In [118], the concept of wisdom of the crowds, introduced in [122], is applied twice to forecast 2010 UK election results using data from social media. Using an ARIMA model they claim to exceed the predictive power of classical surveys.

A quantitative analysis of Tweets is performed in [98] to prove that social media can be a reliable tool about political behavior, applying this technique to competitive races of 2010 and 2012 US congressional elections.

Moreover, in [269], the authors state that volume of tweets is not always enough to capture public opinion and they propose a better but not perfect model able to obtain more accurate results about 2012 American republican presidential election.

Interesting results are obtained observing the bias of pools and Twitter for Donald Trump and Hillary Clinton in 2016 U.S. election, suggesting to not underestimate the effect that different forecasting methods can have on the predictions based on the nature of the method itself (an heterogeneous sample of the voters is not easy to collect) [10].

An improved analysis is performed on Brexit data, classifying through SVM the leave/remain intention of users. In [59], they confirm that this kind of analysis of political topics using social media data can substitute Internet pools and telephone calls, being not only more accurate, but also faster and cheaper.

However, in [126], the limits of Twitter are exposed, revealing the scarce robustness of this approaches. They apply algorithms that obtained good results for one election forecast to other elections, showing that results are not always as good as stated before. They conclude suggesting to investigate impact of different lexicons and the application of machine learning techniques for this task.

Similar analysis has also been performed about German federal election 2009, proving that Twitter can be used as a source to perform political forecasts, since it is widely used for political deliberation and it mirrors the offline political sentiment [292].

An interesting analysis of prediction of political inclination of Twitter users comparing results coming from contents (defined by hashtags used) and networks structure is performed in [73], showing advantages and disadvantages of both the techniques.

Some examples of prediction using syntactic features are the forecast of box-office revenues for movies using tweets about a set of popular movies [13] and the knowledge extraction algorithm proposed in [45, 47] (Sections 3.1 and 3.2).

Sentiment analysis is also one of the most popular techniques, applied to correlate significant events in social, political, cultural and economic sphere with moods extracted from tweets posted in the meantime [33, 220].

Interesting research about political echo chamber must be cited here, where the authors find huge differences between Democrats and Republican behavior on Twitter through also network analysis techniques [70].

5.1.3 Methods

The proposed pipeline can be summarized as follows:

- creation of the dataset (Section 5.1.3);
- selection of appropriate syntactic features (Section 5.1.3);
- selection of a text embedding method (Section 5.1.3);
- selection of a multiclass classifier (Section 5.1.3).

Dataset

The obtained dataset is a collection of Twitter textual data from Italian deputies' accounts.

Firstly, we collect names of the 630 Italian deputies and their corresponding parties from the official website of the Italian parliament[1]. Deputies belonging to small parties are discarded due to their relatively small importance in the actual political situation, obtaining 562 names out of 630 deputies. The four main Italian parties selected are: “MOVIMENTO 5 STELLE", “LEGA - SALVINI PREMIER", “PARTITO DEMOCRATICO" and “FORZA ITALIA - BERLUSCONI PRESIDENTE".

We automatically associate at each deputy his/her official twitter account. Using the twitter API to search for users, the names collected are used as inputs. We often obtain more than one account for the each query, due to homonymy issues. We discard accounts that do not contain in their bio one of the words selected (and their corresponding variations) about politics: 'deputato' (deputy), 'camera' (chamber), 'parlamento' (parliament), 'partito' (party), 'legislatura' (legislature), 'pd', 'lega', 'movimento' (movement), 'stelle' (stars), 'forza italia', 'salvini', 'berlusconi'. Accounts with less than one hundred tweets are also rejected, since our analysis relies on a statistically relevant number of written words to analyze. Finally, if more than one account still corresponds

[1] http://www.camera.it/leg18/1

to a given name of a deputy, the right one is manually selected. This case happened only a couple of times. After the cleaning procedure, 188 twitter accounts corresponding to Italian deputies belonging to one of the four main Italian parties are collected, split as follows: “MOVIMENTO 5 STELLE": 64, “PARTITO DEMOCRATICO": 51, “FORZA ITALIA - BERLUSCONI PRESIDENTE": 39, “LEGA - SALVINI PREMIER": 34. We are aware that this procedure does not find every account belonging to an actual Italian deputy, however we obtained a large enough dataset to perform our analysis. A more accurate analysis can be done by manually searching for politicians' accounts, but we believe that the great part of accounts that we are missing by automatizing the step is not be relevant to the analysis since they are not active enough on Twitter.

For each account found, we collect the last 200 tweets (one Twitter API call) per user u, excluding retweets, and we merge the texts into a single large document d_u. URLs, mentions and every not alphanumerical character are removed to clean the text from non useful features.

The total number of tweets collected is $30k$ because not every account tweeted at least 200 tweets since their registration on the social network. We remark that, for this analysis, no starting date has been selected, since we assume that the political inclination of actual deputies has not changed recently.

Since our hypothesis is that deputies belonging to the same party write in the same way, the large documents obtained should contain enough information to understand the users' political inclination, so to classify accounts into the correct political party.

Selection of syntactic features

To select which kind of syntactic feature is better to classify users, for each user u we tag every word w_u of the document d_u, using a standard tagset [2]. This step is followed by a lemmatization step to reduce inflectional forms of a word to a common base form, performed by the NLP tool *TreeTagger* [262], developed in python and trained using an Italian dataset.

For each user u, from each original document d_u, we obtain five different lists of words:

1. list of every word w_u, ignoring the tags;
2. list of nouns n_u;
3. list of verbs v_u;
4. list of adjectives a_u;
5. list of adverbs ad_u;

[2] http://sslmit.unibo.it/ baroni/collocazioni/itwac.tagset.txt

We perform this step to understand if there is a set of words that improves classification accuracy.

We obtain 40554 different words, 7838 nouns, 2469 verbs, 3118 adjectives and 490 adverbs, discarding what the tagger classifies as "unknown". Other tags are neglected since we think no useful information is contained in those set of words (articles, conjunctions, ...).

Vectorization

A common approach to perform classification tasks expects lists of words w_u (or list of every other feature selected before) embedded into vectors, thus a vectorization step is required.

We try some standard vectorization methods (already described in Chapter 2) to better understand which one is the best embedding technique for our task.

1. *Count Vectorizer* (CV): converts a collection of text documents into a matrix of token counts. Each value represents the number of times that a user u used a word w;
$$CV(w,u) = f_{w,u}$$

2. *Hashing Vectorizer* (HV): converts a collection of text documents into a matrix of token occurrences, using the hashing trick to find the map between the token string name and the feature integer index;

3. *Term Frequency Vectorizer* (TF): converts a collection of text documents into a matrix of term frequencies. Each value represents the frequency that a user u used a word w;
$$TF(w,u) = \frac{f_{w,u}}{\sum_{w' \in d_u} f_{w',u}}$$

4. *Term Frequency - Inverse Document Frequency Vectorizer* (TF-IDF): converts a collection of text documents into a matrix of term frequencies weighted by document frequency;
$$TFIDF(w,u,U) = TF(w,u)IDF(w,U)$$
where U is the set of users,
$$IDF(w,U) = \log \frac{|U|}{|u \in U : w \in d_u|}$$
IDF represents the logarithm of the fraction of the total number of users and the number of users that used the word w.

HV, TF and TF-IDF can be performed with L1 or L2 normalization, obtaining a total of seven different techniques [244]. No stop words are removed during this step.

Classification

Finally, a set of multiclass classifiers is selected to perform the training stage:

1. *Multinomial Logistic Regression*, a generalization of Logistic Regression to Multiclass Problems (four classes), tuning the regularization parameter;

2. *K-neighbors Classifier*, tuning K (the number of neighbors to consider);

3. *Decision Tree*, tuning the depths of the trees;

4. *Random Forest*, tuning depths and number of trees;

5. *Support Vector Classifier*: support vector machines applied for classification tasks, investigating kernel type and appropriate hyper parameters;

6. *MultiLayer Perceptron Classifier*: feed forward fully connected neural network, tuning simple architectural parameters.

For each one of the features selected and vectorization techniques, the classifiers are trained and the results and collected, fine tuning the necessary hyper-parameters with a grid search step.

5.1.4 Evaluation and results

To evaluate the performance of the different methods, we select k-fold cross validation. The dataset is divided in k subsets and each one of them is iteratively selected as test set, while the others are used to train the models. This technique, then, averages the performances to get a more precise evaluation of the model, particularly useful since our dataset consists in only 188 users. We chose $k = 5$ for the whole analysis.

Each approach (a combination of features choice, vectorizer and classifier) goes through a training phase, and finally it is compared with the other approaches. Different metric scores are chosen to obtain accurate insight into the quality of predictions, possibly enabling the observation of biases or other kinds of misclassification issues.

1. $accuracy = \frac{tp+tn}{tp+tn+fp+fn}$

2. $precision = \frac{tp}{tp+fp}$

3. $recall = \frac{tp}{tp+fn}$

4. $f_1 = 2\frac{precision \times recall}{precision+recall}$

where tp is the number of true positives, tn is the number of true negatives, fp is the number of false positives and fn is the number of false negatives. For each party p, true positives are deputies belonging to p and actually predicted correctly, false positives are deputies wrongly predicted to belong to p, etc.

Since this is a multiclass classification problem, precision, recall and f1 score are different for each class, and the final value is the average, considering the unbalancement of the number of politicians per party.

Results

Here we compare different approaches using average accuracy on 5-fold cross validation.

The highest value of accuracy is obtained using only nouns, vectorized with TF-IDF (L2 norm). Both Multinomial Logistic Regression and simple Multilayer Perceptron Classifier obtain an accuracy of 0.89, with similar values of precision, recall and f1 score (Table 5.2).

Table 5.2: *Evaluation metrics for the five best methods (features-vectorizer-classifier combinations) averaged over the four parties considered.*

features	vectorizer	accuracy	precision	recall	f1 score
nouns	tf-idf L2	0.89	0.91	0.87	0.87
every word	tf L2	0.86	0.86	0.84	0.84
every word	hv L2	0.86	0.87	0.84	0.84
every word	tf-idf L2	0.86	0.87	0.84	0.84
nouns	cv	0.85	0.85	0.84	0.84

Similar but lower accuracies (0.86) are obtained using every tweeted word, using both Hashing Vectorizer, Term Frequency Vectorizer and TF-IDF with L2 norm. Thus, we can state that cleaning the tweets keeping just nouns increases the performance of the classification. Features like adjectives, verbs or adverbs obtain at best an accuracy respectively of 0.75, 0.65 and 0.50, meaning that they do not contain enough information to perform this kind of classification. These words are not used in a different way by politicians belonging to different parties, on the contrary to nouns.

As expected, TF-IDF is the best vectorizer since it can weight words taking into consideration also if they appear in tweets of other deputies, giving more importance to specific words and penalizing more common words.

K-Neighbors Classifier, Decision Trees, Random Forest and SVC do not perform well enough, obtaining very low scores for every vectorizer and features selected. Probably a more rigorous fine tuning of parameters can lead to better results, but it is out of the scope of this section. This analysis proves that politicians belonging to the same party tend to write in the same way. Precisely, the main feature that differentiate between parties are the nouns used. It is important to take into account also the presence of words in other tweets to perform analysis (using TF-IDF vectorizer), and a simple Multinomial Logistic Regression can be trained to obtain good results. We prefer to use the latter classifier since the algorithm is more easily interpretable with respect to a Multilayer Perceptron, with no loss of precision.

5.1.5 Further analysis

After finding a good processing pipeline (method) that can classify politicians given their tweeted texts, we continue our analysis inspecting the gathered dataset.

Firstly we perform a TF-IDF transformation with Euclidean norm on the whole dataset of politicians nouns, to obtain a set of vectors in a $8k$ dimensional space.

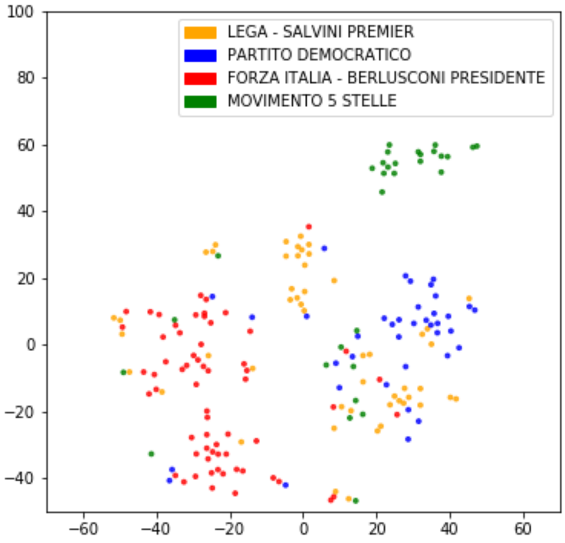
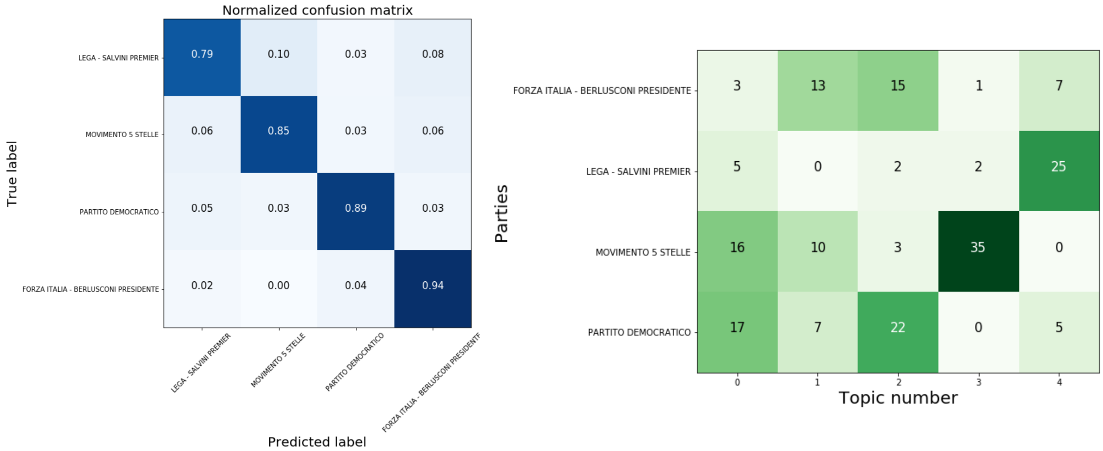


Figure 5.1: *t-SNE 2d projection of TF-IDF vectors calculated using nouns, where each color represents a different party.*

Figure 5.1 shows the projection of the highly dimensional vectors into a 2 dimensional space using t-distributed stochastic neighbor embedding (t-SNE) as a visualization technique [297]. We can easily notice how three out of four parties are very well defined, while politicians belonging “LEGA - SALVINI PREMIER" are largely spread. This suggests that members of that party does not have a specific dictionary of “preferred” words, as the other parties do.

We can verify this hypothesis looking at Figure 5.2a, a normalized confusion matrix that shows insights on the misclassification errors. The party with fewer true positives is in fact “LEGA - SALVINI PREMIER", which true deputies are often classified as belonging to “MOVIMENTO 5 STELLE" (0.10), or as belonging to “FORZA ITALIA



(a)

(b)

Figure 5.2: *Left: Confusion matrix for the predictions of the best classifier (nouns with TF-IDF L2 norm, Logistic Regression). The values reported are the mean with relation to the 5-fold cross validation results; Right: Automatic detection of 5 topics from the corpus of tweets by political party.*

- BERLUSCONI PRESIDENTE" (0.08), suggesting some syntactic similarity between these parties.

However, it is also important to notice that the silhouette score [256] applied to the dataset vectorized with TF-IDF with L2 norm, has a low value (0.01), indicating that this kind of high dimensional vectors does not form compact and separate clusters, since, of course, users are not using a completely different sets of nouns. The great part of nouns used is shared with other parties, while just a few terms are decisive for classification purposes.

We now shift our focus on specific nouns in the tweets. In Table A.4 in Appendix A, nouns, whose coefficient of Multinomial Logistic Regression is higher/lower, are listed for every party. For example, the word “centrodestra" (centre-right) is the most significant noun for users belonging to “FORZA ITALIA - BERLUSCONI PRESIDENTE", meaning that this term is decisive during the classification of a user to than party, while the noun “lega", being in the last place, will have the opposite role in the prediction step. As expected, nouns of the parties, like “movimento" (movement) and “stella" (star) for “MOVIMENTO 5 STELLE" are in the first positions, while they are in the last positions for others parties, suggesting that politicians tend to talk mostly about their own parties. Interesting is also the presence of nouns like “cittadino" (citizen) and “gente" (people) for populist parties in the first positions, while for other parties they are in last positions. Finally, words like “nord" (north) and “sud" (south) can suggest a particular focus of the selected party with relation to the Italian geographical region, obtaining a hint on where the political interests of the parties are.

Finally, we apply a topic detection algorithm to the data to get further insights into what the tweets are about. We selected LSA method [105], approximately decomposing the TD-IDF matrix X (number of deputies times number of nouns) obtained before into the product of three matrices, U (number of deputies times number of topics), S (a diagonal matrix with sorted eigenvalues) and V (number of nouns times number of topics). The S matrix describes how much the topics are important, while U contains information on how the deputies are related to the topics, and V groups the nouns into different topics. Thus, observing this decomposition we obtain information about how different parties are related to different interests.

We chose a number of topics of five, and we decompose the TF-IDF matrix as described above. Inspecting matrix U, we can chose the most relevant topic per deputy. In Figure 5.2b we show the results. Interesting how topic 4 is dominated by “LEGA - SALVINI PREMIER", while “MOVIMENTO 5 STELLE" is more focused on topic 3. The other two parties are more balanced between 2 topics. Analyzing which nouns characterize the topics through the matrix V we notice that topic 4 is composed of “moschea" (mosque), “immigrato" (immigrant), “festa" (party), “gazebo" (gazebo), while topic 3 by “cittadino" (citizen), “video" (video), “appuntamento" (appointment), reflecting, as expected, the political inclination of those parties. Of course most of the words that characterize the topics are related to politics, such as “legge" (law), “camera" (chamber), “ministro" (minister), “governo" (government), since the main topic of the shared tweets is politics, but we are still able to identify subtopics highly related to the most characterizing ideas of the parties.

5.1.6 Conclusion

In this section, I investigated how to apply classical Natural Language Processing to obtain insights into the political situation in Italy. Since the only requirement is a language-specific word tagger, I can repeat the same analysis for any country whose people use a text-based social network for political discussions and propaganda. Results prove our hypothesis: deputies belonging to the same party use the same or similar words (in particular, nouns) when tweeting. I use this to classify accounts obtaining good performances once I select the best vectorization technique.

This work focuses on deputies due to the ease to collect a ground truth. More challenging analyses involve citizens whose political inclination is unknown and changes more frequently. Thus I expect this task to be harder to perform and evaluate.

In the next section, I investigate the 2020 Italian Referendum focusing on citizens and not politicians. To solve the missing ground truth problem, I shift the task from user classification to single tweets classification. I define Hashtag-based semi-automatic labelling to obtain the ground truth. I use more advanced text classification techniques, previously described in detail in Chapter 2, due to the increased difficulty of the task.

5.2 Content-based Stance Classification of Tweets about the 2020 Italian Constitutional Referendum[1]

5.2.1 Introduction

On September 20 and 21, 2020, a constitutional referendum was held in Italy to reduce the number of parliamentarians (from 630 to 400). 69.96% of the voters approved it, with a voter turnout of about 51%[3]. Since the main Italian political parties supported the referendum, at first the outcome was obvious, but, through a huge activity on social media, opposers unsuccessfully tried to overturn the result. The referendum was a *confirmatory* referendum: voters were asked to approve a law. Thus, we refer to people that voted "yes", agreeing with the introduction of the new law that reduces the number of parliamentarians, as Supporters, and we refer to people that voted "no", against the introduction of the new law, as Opposers.

Since an always greater number of people share their thoughts online, social network analysis helps understanding the causes and forecasting the outcomes of political events, in parallel with already widely used approaches such as surveys and pools [54]. Like surveys, *selection biases* are hard to remove. Social media users and citizens have different demographic distributions, resulting in under-represented categories of people (e.g., elderly people) [205][4]. Moreover, social media are also populated by bots, softwares that run accounts and automatically share content, introducing noise and bias in the collected data [115]. These accounts are not run by real people and the data shared by them should not be included to perform analysis and statistics. However, a big advantage of the analysis of social media data is the higher magnitude of available data, easy to collect and process. It is often less expensive to collect content from social media than using classical approaches.

In this study we collect and analyze Twitter data about the Italian referendum in 2020. Our contributions can be summarized as follows:

- We collect and publicly share a corpus of $1.2M$ tweets about the Italian referendum in 2020. This is a rare and fundamental resource for NLP analysis, expecially stance detection, for non-English texts[5];

- We design a content-based, semi-automatic, approach to label big magnitudes of textual data through hashtags. We obtain a set of $85k$ cleaned labeled texts with low human effort;

[1] Authors: Marco Di Giovanni, first author, conceptualization, implementation, experiment design, writing; Marco Brambilla, conceptualization, editing. [94]

[3] https://en.wikipedia.org/wiki/2020_Italian_constitutional_referendum

[4] https://www.statista.com/statistics/283119/age-distribution-of-global-twitter-users/

[5] The dataset is publicly available at https://github.com/marco-digio/italian-referendum-2020

- We fine-tune an accurate text classifier to detect the stance of tweets (Support or Against the referendum). We also successfully apply it to classify tweets that the semi-automatic approach *cannot* label;

- We inspect three common text biases (length-bias, lexical-bias and sentiment-bias), observing that our dataset does not suffer from them;

- We discuss the discrepancy between the collected data from Twitter and the real outcome of the referendum, including possible further investigation essential to understand the phenomenon.

5.2.2 Related Works

Numerous published works correlate social media data with elections or referendums. The main and most studied recent event is the Brexit referendum, largely investigated from many different points of view [59, 89, 136, 156, 193, 211], but many other political events have been analyzed from a social media perspective [82, 87, 234, 275, 292].

A general approach to quantify controversy in social media has been proposed by [125], designing a graph-based approach using solely on the underneath social graphs. This approach is language independent, relying solely on the social structure of communities of users, but computational expensive. Another approach has been proposed, that includes the content of texts to make more precise and fast computations [84].

We investigate this event from a content-based *stance detection* perspective [174], analyzing only user-generated content to detect the inclination about the referendum in Italy. There are few works about stance detection with non-English tweets [295]. [181] collect a similar dataset for the Italian referendum in 2016. They tackle the stance detection task by adding to simple NLP approaches, such as bag of hashtags, bag of mentions or bag of replies, network based features obtained by clustering the retweet/quote/reply networks with Louvain Modularity algorithm. They also analyze the datasets from a diachronic perspective by splitting the time window into four sections based on the dates of referendum-related events. Other works focus on the Italian political situation of Twitter users with content-based approaches [96, 246, 247]. They collect tweets shared by politicians and their followers, and train accurate classifiers that predict the political inclination of users, without considering the social interactions: the content shared contains enough information to successfully perform classification of political inclination.

Similar tasks have been proposed at SemEval 2016 [209], IberEval 2017 [284], IberEval 2018 [285] and finally at EVALITA 2020 [66], where teams were challenged to detect stances of manually labeled Italian Tweets about the Sardine Movement. We remark the difficulty of such tasks by looking at the performance of the best team [131], that fine-tuned an Italian pre-trained BERT model [91] and augmented the data with results from three auxiliary tasks.

A comparative study [127] shows that for stance-detection datasets of English texts from Web and Social Media, BERT model achieves the best performance, but there is still much room for improvements.

5.2.3 Data Collection, Description and Labeling

The dataset is collected from **Twitter**[6], a micro-blogging platform widely used to discuss trending topics, whose official API allows a fast and comprehensive implementation. On Twitter, users share *tweets*, small texts (up to 280 characters) that can be enriched with images, videos or URLs. Other users can *quote* (or *retweet*) another tweet by sharing it with (or without) a personal comment. A user can also *follow* other users to get a notification when they tweet (retweet or quote), and can be followed by other users.

We query data about the referendum held in Italy in September 2020 by searching Italian tweets, containing at least one of the keywords reported in Table 5.3, usually used as hashtags, but not always. In total we collected $1.2M$ Italian tweets posted between 01/08/2020 and 01/10/2020 by about $111k$ users.

Table 5.3: *List of keywords used to filter relevant tweets. They refer to vote, parliamentarians, cuts and referendum. We substitute * with no, si and sì (yes in Italian).*

iovoto*	parlamentari	iovoto*taglioparlamentari
voto*	vota_efaivotare*	tagliodeiparlamentari
vota*	referendum	referendum2020_iovoto*
votare*	referendum2020	iovoto*_referendum2020
unitiperil*	maratonaperil*	cittadiniperil*

The keywords are refined and validated iteratively. Starting from three keywords (referendum, iovotosì - IVoteYes, iovotono - IVoteNo), we inspect the most frequent hashtags and, if related to the topic, we add them to the query. In Figure 5.3 we show the most used hashtags in our complete dataset. Many frequent hashtags have no clear and safe connection with the referendum, thus we do not select them as keywords during the collection step, such surnames of politicians ("dimaio") and political parties ("m5s").

Hashtag-based Semi-automatic Labeling

Manually labeling big data sets is an expensive and not-scalable approach. Usually more than one annotator, fluent in the selected language, is required to produce a reli-

[6] https://twitter.com

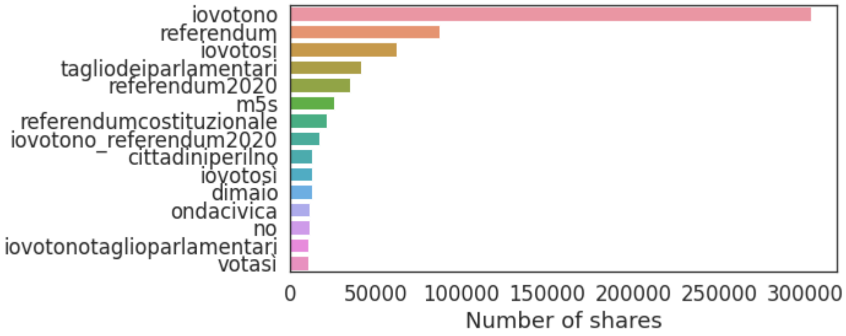


Figure 5.3: *Mostly shared hashtags in the dataset.*

able label, and the time and cost to obtain a data set large enough to train an accurate classifier is usually high.

Graph-based approaches have obtained impressive results when applied to detect stances in controversial debates [75, 125]. These approaches are mainly used to label user by looking at the nearest community in the social graph. They firstly define the graph structure, e.g. retweet graph, and then they apply community detection algorithms to partition the bigger connected component of the graph.

We design a content-based approach to semi-automatically label large sets of tweets. Different from the graph-based approaches, we label *single* tweets, while the graph approaches work at the user-level. The approach is based on *hashtags*, often used to express the inclination of users about a topic [209]. Trending hashtags attract audience and get the attention of other users in the social network[7].

We pick two main classes: in *Support* of the referendum and *Against* the referendum. We define as *Gold hashtags* the hashtags that clearly state a side in the vaccine debate. We plan to collect two sets of Gold hashtags, one for each side of the debate. If a tweet contains at least one of the Gold hashtags, we define its stance as the stance of the hashtag. Tweets containing at least one Gold hashtag from both sides are discarded. Firstly, we select two Gold hashtags, one for each side: #iovotosì (I Vote Yes) for the Support class and #iovotono (I Vote No) for the Against class. Note that in Italian the word *yes* is translated as *sì*, with the grave accent that is often omitted in informal texts, such as tweets. Thus, in the whole paper, every time we refer to the word *sì*, we include also the word *si*, without the accent. Two annotators manually validate this initial selection by inspecting 100 tweets for each class and finding only 4 tweets that clearly belongs to the opposite stance. They were used to attract the attention of the other side or to delegitimise a specific hashtag., e.g. “I cannot understand people that write #IVoteYes". However, our validation process confirms that these tweets are rare and introduce little noise to the data set.

[7] Twitter has a specific section for trending hashtags and keywords
https://twitter.com/explore/tabs/trending

We iteratively add new hashtags by inspecting the most frequent co-occurring ones and manually selecting the most pertinent ones, basing the selection on their meaning. An example of discarded hashtags is #conte (the surname of the Prime Minister of Italy at the time of the Referendum), highly co-occuring with #iovotono, since we cannot safely assume that it was used only by users Against the referendum. We also discard hashtags that co-occur with hashtags from both sides in similar percentages. An example is #referendum, obviously frequently used by both sides of the debate. Finally, after each iteration two annotators manually validate the selected hashtags, as previously described for the initial Gold hashtags. An hashtag passes the validation if the percentage of tweets that is classified by at least one annotator as belonging to the opposite class is lower than 10%. We finally obtain two final sets of **Support Gold hashtags** and **Against Gold hashtags**, that allows us to get about $450k$ labeled tweets by manually labeling *few hundreds*. The selected Gold hashtags are the keywords reported in Table 5.3 that contain the * symbol. The symbol is substituted with the corresponding stance (“sì" or “no"). For example, #referendum2020_iovotono is a Gold hashtag for Against class, while #referendum2020_iovotosì (and #referendum2020_iovotosi) is a Gold hashtag for Support class. Since no other hashtag among the 50 most-frequent ones passes the full validation procedure, we end the labeling phase.

Note that we label tweets containing at least one hashtag from a single set in the corresponding class, while tweets with at least one hashtag from both sets as Both and tweets without any hashtag from both sets as Unknown. We remark that Both and Unknown tweets cannot be safely considered *neutral* since they can express a stance without explicitly using one of the selected hashtags, or using both of them (Table 5.4 reports an example of a neutral tweet labeled as *Both* (A) and a Support tweet labeled as *Both* (B)). This is the main limitation of this semi-automatic labeling procedure: no neutral class can be safely defined, thus we can only train a binary-classifier, leaving for future works the design of a three-classes stance detector.

We label *retweets* by looking at the hashtags in the original tweet, we label *quotes* by only looking at the hashtags in the quote itself, not at the quoted hashtags. In Table 5.5 we report the statistics of the obtained labeled dataset. Original tweets are tweets that are neither retweets nor quotes of other tweets, nor replies to other tweets.

Table 5.4: *Translated examples of tweets containing both the Gold hashtag #iovoto and #iovotosì. (A) shows a neutral tweet, (B) shows a Supporter attacking the point of view of people Against the referendum.*

	Tweets using both #IoVotoSì and #IoVotoNo
A	In a few days we will meet at the ballot boxes to express our preference about the #CutOfParliamentarians. While waiting, let's retrace the most famous referendums in the history of the Republic. #Referendum2020 #IVoteYes #IVoteNo
B	Let's dismantle some lies about #IVoteNO. The #CutOfParliamentarians is a reform that fixes the Italian distortion of having a very big number of elected people. Who talks about dictatorship is only using the usual fear strategy to keep a useless privilege. #IVoteYes

Table 5.5: *Tweets Statistics.*

Label	**Tweets**	**Original**	**Retweets**	**Quotes**	**Replies**
Support	93149	74086	2890	10572	5665
Against	364865	291185	15368	34559	24145
Both	4224	2796	145	246	1042
Unknown	353033	236743	16600	53119	47059
Total	815271	604810	35003	98496	77911

Temporal Analysis

In Figure 5.4 (top) we show the distribution of tweets, grouped by their stance, during the time window selected, highlighting the referendum day. We notice a first peak around the August 8, due to an unrelated event about parliamentarians, that we accidentally included, since we used *parliamentarians* as a keyword to filter tweets. To remove noise and unrelated data, we discard all tweets posted before August 15 in the following analyses.

We also notice a huge peak of Unknown tweets during the referendum days, probably because users switched from the old hashtags *#IVoteYes* and *#IVoteNo* to their past tense versions (*#IVotedYes* and *#IVotedNo*). Thus, we discard tweets posted after September 19. Moreover, we do not want to influence our stance classification with tweets posted after the referendum.

In Figure 5.4 (bottom) we show how the ratio between Support and Against tweets evolves during the time window, observing constant values around 0.25 from August 15 to September 19. Thus, the daily number of tweets Against the referendum is four times bigger than the number of tweets Supporting it, further confirmed in Table 5.5, where the total number of Support tweets is four times smaller than the total number of tweets Against the referendum. We also notice big peaks and valleys outside the

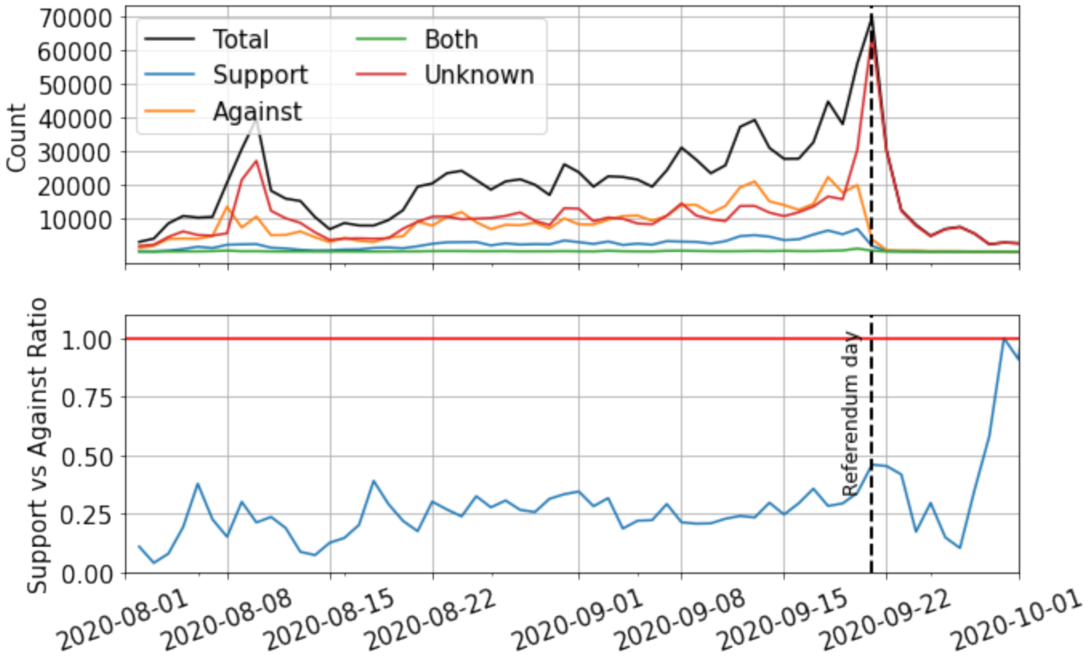


Figure 5.4: *Top: Number of daily shared tweets, grouped by stance. Bottom: Daily Support vs Against Ratio. The higher the ratio, the greater the number of tweet Against the referendum. The red line (1) sets the value of equal number of Support and Against tweets.*

selected time window, caused by the low number of daily posted tweets.

5.2.4 Data Analysis

In this section we describe the cleaning process, the stance classifiers and their results on the collected dataset.

Data Cleaning

Before training a stance classifier, we clean the text of tweets through the following procedure.

Texts are lowercased, URLs are removed and spaces are standardized. **We remove Gold hashtags** (see Table 5.3) since they were used to automatically label tweets and users, thus maintaining them will introduce a strong bias in the trained models. We keep the other hashtags since they could encode useful information and are not a clear source of bias. Tweets containing at least half of the characters as hashtags are also removed, since they are too noisy. They are usually used by bots to collect the daily trending hashtags. To prevent overfitting we remove duplicate texts, including retweets. We also remove texts shorter than 20 characters, that usually comment URLs or other tweets, being difficult to understand and contextualize. We keep emoji as they include useful information, e.g., the scissor emoji was mainly used by Supporters of the referendum since they want to *cut* the number of parliamentarians. We select only tweets shared after 15/08/2020 and before 20/09/2020, the first referendum day.

Stance classification

We analyze the dataset from a stance classification perspective.

Due to the impossibility to interpret the tweets labeled as Both or Unknown, we formulate the tweet stance classification task as a binary classification problem: the two classes represent tweets Supporting or Against the referendum. We obtain an unbalanced clean datasets: $85k$ tweets, of which 80% Against the referendum. To obtain a balanced dataset, over-sampling the *Support* class leads to slightly better results in the Validation dataset, but worse results on the Test set, probably due to overfitting, while under-sampling the *Against* class leads to worse results due to the removal of 60% of the original dataset.

We select three models (one baseline and two commonly used architectures):

- Majority classifier (Baseline);

- FastText [165], a fast approach widely used for text classification. Its architecture is similar to the CBOW model in Word2Vec [203]: a look-up table of words is used to generate word representations, that are averaged and fed into a linear classifier. A softmax function is used to compute the probability distribution over the classes. To include the local order of words, n-grams are used as additional features, with the *hashing trick* to keep the approach fast and memory efficient. FastText is known to reach performances on par with some deep learning methods, while being much faster (for further details see Section 2.3.2);

- BERT [91], a Transformer-based model [298] that reaches state-of-the-art performances on many heterogeneous benchmark tasks. The model is pre-trained on large corpora of unsupervised texts using two self-supervised techniques: Masked Language Models (MLM) task and Next Sentence Prediction (NSP) task. Pre-trained weights are available on the Huggingface models repository [313]. We select a model pre-trained on a concatenation of Italian Wikipedia texts, OPUS corpora [288] and OSCAR corpus [221], performed by MDZ Digital Library[8]. We fine-tune the model on our data[9] (for further details see Section 2.4.4).

Results

In Table 5.6 (left) we report the results of a 5-fold cross validation process. We select Area Under the ROC curve [112], weighted F1-score (the F1 score for the classes are

[8] https://huggingface.co/dbmdz/bert-base-italian-xxl-uncased

[9] Fine-tuning performed on a single NVIDIA Tesla P100, for 5 epochs. Best weights selected by minimizing the evaluation loss. Learning rate (10^{-5}) set through grid search.

Table 5.6: *Area under ROC (AUROC), weighted F1 score ($F1_w$) and F1 score of the Supporters ($F1_s$) of the three models, as 5-fold Cross Validation on the training set (left) and on the Test Sets of* 227 *randomly selected and manually evaluated texts.*

	Validation			Test		
Model	AUROC	$F1_w$	$F1_s$	AUROC	$F1_w$	$F1_s$
Baseline	0.50	0.78	0	0.50	0.52	0
FastText	0.74	**0.89**	0.56	0.65	0.59	0.18
BERT	**0.88**	0.86	**0.63**	**0.78**	**0.71**	**0.5**

weighted by the support, i.e., the number of true instances for each class) and $F1_s$, the F1 score on the Support class (the under-represented class, that, by definition, a Majority classifier cannot detect).

Both FastText model and BERT outperform the Random Baseline approach, the latter obtaining higher AUROC and $F1_s$.

However, our goal is to predict the stance of tweets that do *not* share a Gold Hashtag. We use these models, trained on the big dataset labeled using Gold hashtags, to predict tweets that do not contain Gold Hashtags, thus tweets that, with the previously described automatic approach, were labeled as Unknown. Two human annotators manually labeled 500 randomly sampled tweets. After removing neutral and incomprehensible texts, we obtain a dataset of 227 tweets, of which 78 labeled as Supporters. We test our models on this dataset, the results are reported in Table 5.6 (right), confirming that even if there is a gap among the Validation performances and the Test performances, BERT did not strongly overfit the Training data.

Finally, we obtain an approximate statistic of the total number of tweets Supporting and Against the referendum by predicting the stance of every tweet previously labeled as Unknown ($110k$ tweets). It results in about 20% of Unknown tweets classified as Supporters, confirming the general number of tweets Against the referendum is four times bigger that the number of shared tweets Supporting it. However, we cannot validate this result since we do not have manually labeled the full dataset.

5.2.5 Biases analysis

In this section we inspect three common biases that often affect the accuracies of classifiers: Length of texts, Lexicon and Sentiment.

Length Analysis

The length of sentences, defined as the number of characters or tokens, often influences the prediction of a model, acting as a bias. In Figure 5.5 we plot the distribution of lengths of tweets calculated as the number of characters, after the cleaning procedure (there are no tweets shorter than 20 characters). There is no evident difference between

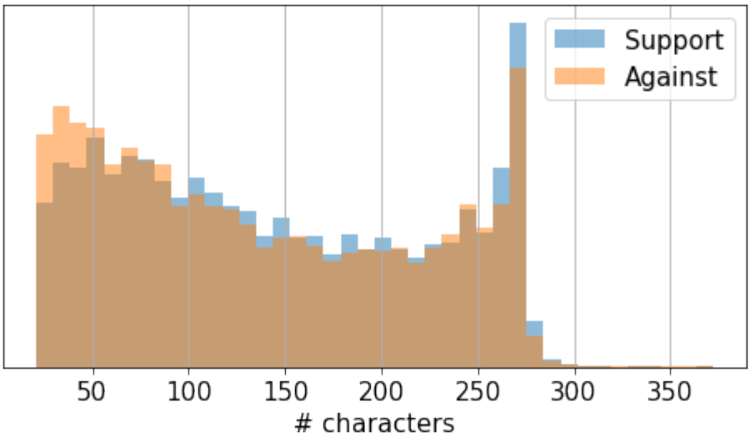


Figure 5.5: *Length distribution of generated tweets grouped by stance. There is no significant difference in the normalized distributions.*

the distribution of the number of characters in tweets labeled as Support or Against, suggesting that no length-bias is present in our dataset.

Lexicon analysis

We check if tweets in different stances use similar lexicons. A big lexicon overlap in the dataset results in an accurate classifier that must learn the *meaning* of sentences, while a small lexicon overlap in the dataset allows the detection of specific words to be sufficient to make a prediction, neglecting the real meaning of the texts. We quantify the lexicon difference by computing the Pointwise Mutual Information (PMI) between words and classes [146].

A high PMI score of a word in a class is obtained when the word is used mainly in tweets belonging to that class. For this analysis, we discard Italian stop words collected from the NLTK library [28].

We report in Table 5.7 the first five words for each class, sorted by PMI score and the proportion of texts in each class containing each word. The frequency of words with higher PMI is low, thus we conclude that the two stances use mostly similar lexicons. A classifier cannot safely rely on the presence of specific words since the most indicative ones (higher PMI score) are not frequent enough. For example, the most frequent word among the top-5 is orgoglio5stelle, a keyword used by Supporters of the Referendum stating that they are proud of their party (5 stars) because the referendum was held by them. However, only 3% of the Supporter texts include this word.

Table 5.7: *Top 5 tokens ranked by PMI (Pointwise Mutual Information) scores and the proportion of texts in each class containing each word.*

Support	%	Against	%
orgoglio5stelle	3.0	ondacivica	2.2
scissors emoji	0.3	30giorni_iovotono	0.5
laricchiapresidente	0.9	iostoconsalvini	0.5
pugliafutura	0.5	noino	0.4
rotolidistampaigienica	0.3	darevocealreferendum	0.4

5.2.6 Sentiment analysis

We distinguish between sentiment classification and stance classification by searching for a correlation between sentiment and stance in the datasets. Our goal is to have a stance classifier that does not rely on the sentiment of tweets to make a prediction. If Support and Against tweets are unbalanced in the Positive and Negative sentiment classes, the dataset contains a sentiment-bias.

We compute the sentiment scores of tweets and users using Neuraly's "Bert-italian-cased-sentiment" model[10] hosted by Huggingface [313]. It is a BERT base model trained from an instance of "bert-base-italian-cased"[11] and fine-tuned on an Italian dataset of $45k$ tweets on a 3-classes sentiment analysis task (negative, neutral and positive) from SENTIPOLC task at EVALITA 2016 [17], obtaining 82% test accuracy.

In Figure 5.6 we show the Kernel Density Estimation plot of positive and negative sentiment of tweets grouped by stance. The probability of being neutral is not shown as it can be obtained with $1 - p('positive') - p('negative')$. Since the distributions of the sentiments largely overlap, we conclude that there is no sentiment-bias in our datasets. It is further confirmed by looking at the actual predictions: for both Support and Against texts, 63% of them are classified as Negative, 25% as Neutral and 15% as Positive.

5.2.7 Discussion

Discrepancy between Twitter activity and the Referendum outcome

We notice a huge discrepancy between what users posted on Twitter and what citizens voted. The fraction of tweets and users that explicitly state their stance (and our prediction of tweets and users that do not) is very different from the final outcome of the referendum (69.96% of the voters approved it): the number of tweets with a Gold Hashtag Against the referendum is 4 times higher than the number of tweets with a Supporter Gold Hashtag, and the number of Unknown tweets that our best classifier

[10] https://huggingface.co/neuraly/bert-base-italian-cased-sentiment

[11] https://huggingface.co/dbmdz/bert-base-italian-cased

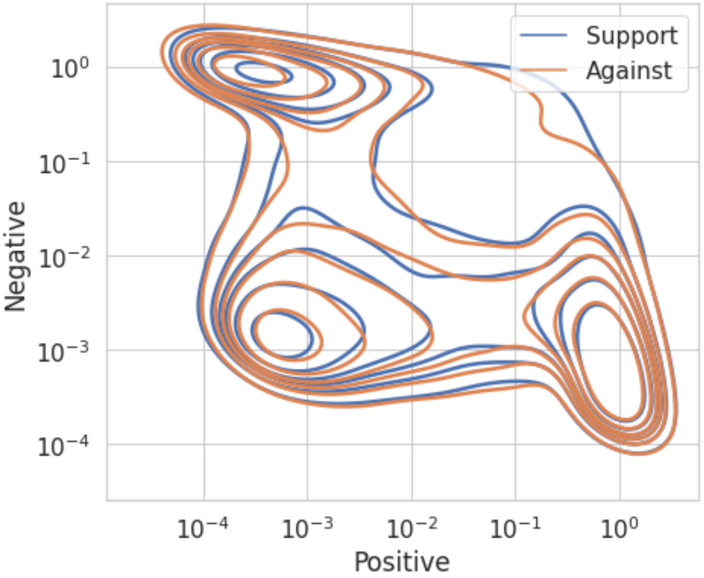


Figure 5.6: *Sentiment distribution of generated tweets grouped by stance. There is no evident difference in the distributions. To improve the visualization, we use the same number of data points for both stances, downsampling the texts Against the referendum.*

predicts as Support or Against the referendum follows the same proportion. By looking only at what is shared online, we could have easily guessed that the Opposers won the referendum, while the real outcome is the opposite.

To further understand this discrepancy, we briefly inspect the differences in social characteristics of users. We label users as Support (Against) if they share only tweets previously labeled as Support (Against) the referendum. Figure 5.7 shows the normalized distribution of number of *followers* and number of *following* of users Supporting and Against the referendum. No difference in shape proves that the social audience of the two sides of users is quantitatively similar (the tails of the figures are cut for visualization purposes). Inspecting the most followed and following users (long tail of the distribution), we notice that among the top-10, exactly half of them are Supporters and half are Against the referendum, confirming our finding. Thus we conclude that Supporters won the referendum, not because they tweeted more than Opposers (they actually tweeted 4 times less than the people against the referendum), neither because they have more audience (the distributions of number of followers and following people is similar). We leave for future works the inspection of more detailed graph-related quantities, such as centrality of users in the network and topological measures to describe the graph structure.

We observed an event where the majority of voters were silent, or not even present on Social Media, while the minority was loud. This phenomenon implies not only that restricting the focus on social media to fully analyze an event could lead to extremely wrong forecasts, but also that the user perception of the general political situation can

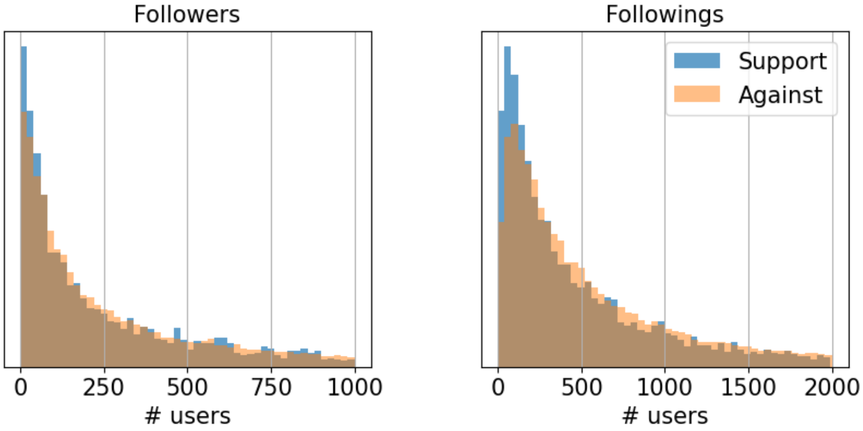


Figure 5.7: *Distribution of followers (left) and following (right) users of users Supporting and Against the referendum.*

be influenced by an unrealistic image of the public opinion on social media that does not match the real sentiment towards the topic.

Ethical Considerations

Political inclinations of people is a sensitive topic. This work is meant to be a exploration on how to apply state-of-the-art NLP techniques to predict the stance of tweets about a political event, and whether they can help to perform more accurate forecasts of the outcome of a political event. Due to privacy issues, we do not share the trained model nor the obtained labels of tweets. However, we share the dehydrated collected tweets and the set of keywords to obtain the gold labels. These data allow researchers to reproduce the results but do not contain sensitive information, meeting the Twitter's Terms of Service[12]. In this study we prove that the political inclination of users can be detected by modern NLP approaches, *even if no evident hashtags of keywords are shared in a tweet.* Thus, we suggest a thoughtful and appropriate usage of social networks in order to keep private sensitive information.

5.2.8 Conclusion

Thanks to the last referendum in Italy, we collected an Italian stance detection user-generated dataset. The dataset consists of $1.2M$ tweets, of which $85k$ are cleaned and labelled as Supporters or Against the referendum. The designed hashtag-based semi-automatic labelling approach allows us to train an accurate classifier that generalizes well also on tweets that do not contain Gold hashtags. We considered three common dataset biases (length-bias, lexicon-bias and sentiment-bias), confirming no significant dangers. Finally, we investigated the discrepancy between the fraction of collected

[12] https://twitter.com/en/privacy

tweets labelled by stance and the referendum outcome. Based on our findings, we suggest that data from social media does not globally reflect citizens ideas, and forecasts should consider this before performing predictions.

In future works, we plan to build a three-classes stance classifier that can also predict neutral texts. We observed magnitudes of tweets that do not explicitly state a stance but still contain useful information. We will also move the focus from tweets to users, detecting their inclination by looking at the history of shared tweets. We believe that the investigation of users that changed their stance during the time window could help us understand how social media influences people. Finally, we observe that our classifier does not generalize well on other Italian stance-detection datasets due to the high specificity of the task: the model learned the debate about the 2020 Italian constitutional referendum and its actors' inclination, but the knowledge obtained is not adequate to perform a zero-shot transfer to other data sets. However, we plan to investigate if we can gain boosts of performances in a multi-task and multi-source context, training a model on multiple similar tasks and data.

In this Chapter, I have collected two works about NLP techniques applied to textual content about politics shared on Social Media. The results are promising and suggest that similar and more recent approaches could be helpful for future political events. These analyses are possible if the participation of Italian citizens on Twitter and other Social Networking sites does not decrease. However, these works also suggest that it is easy to detect political inclinations of citizens by investigating the publicly available content shared on Social Media. There should be no need to specify that the best way to keep delicate personal information private is to limit the use of Social Media as the user profiling techniques evolve.

CHAPTER 6

Applications: COVID-19

In this Chapter, I report three published works about investigations of NLP techniques applied to CODIV-19 discussions and, in particular, about vaccines against the virus, on Social Media.

The first study is a general investigation of Facebook posts about COVID-19, focusing on three misinformation topics: the relationship between migrants and the virus, the origin of the virus in a laboratory and the connection between 5G and the virus. I report here parts, designed and implemented by me, of the complete work [139] where I apply text analysis techniques to obtain further insights into the situation. The rest of the original work reports network analysis and misinformation sources, not included in this thesis.

The second study is the presentation of Vaccinitaly, a project to monitor Italian conversations around vaccines on multiple social media (Twitter, Facebook) to understand the interplay between online public discourse and the vaccine roll-out campaign in Italy. The main focus of the study is misinformation, with a detailed description of the sources of URLs shared on the platforms. We also link online conversations with geographical details on the ongoing vaccination campaign, e.g. the number of doses administered in each Italian region.

Finally, the third study shows the design of a stance classifier of tweets. Using the previously described dataset, I focus on the textual content of collected tweets and, with the help of hashtag-based semi-automatic ground truth, I train a Transformer-based model to predict whether a tweet has a pro-vax or no-vax stance.

Vaccinitaly is an ongoing project that we are now expanding to include many European countries. We plan to also extend the textual analyses with multilingual approaches to obtain a broader view of the phenomenon.

6.1 Information disorders on Italian Facebook during COVID-19 infodemic[1]

6.1.1 Introduction and related work

The spread of a novel coronavirus (COVID-19) in the past months has changed in an unprecedented way the everyday life of people on a global scale. According to World Health Organization (WHO), at the time of this writing (August 2020) the pandemic has caused over 23 M confirmed cases, with more than 809 k fatalities globally speaking[2]. Italy, in particular, has been one of the first European countries to be severely hit by the pandemic, as the virus spread outside China borders at the end of January, and to implement national lockdown on the 8th of March [35,102]. Following Italy and China, national lockdowns have been adopted by most countries around the world, drastically reducing mobility flows in order to circumvent the spread [120].

In relation to the emergency, the term "infodemic" has been coined to describe the risks related to the massive spread of harmful and malicious content on online social platforms [323], as misinformation could support the spread of the virus undermining medical efforts and, at the same time, drive societal mistrust producing other direct damages [323]. In response, several contemporary works have provided different perspectives on this phenomenon. Authors of [121] analyzed more than 100 millions Twitter messages posted worldwide in 64 languages and found correspondence between waves of unreliable and low-quality information and the epidemic ones. Authors of [318] have investigated the prevalence of low-credibility content in relation to the activity of social bots, showing that the combined amount of unreliable information is comparable to the retweets of articles published on The New York Times alone. Finally, authors of [68] have carried out a comparative analysis of information diffusion on different social platforms, from Twitter to Reddit, finding different volumes of misinformation in different environments.

As a matter of fact, ever since 2016 US presidential elections we observed a growing concern of the research community over deceptive information spreading on online social networks [5,184,235,266]. In Italy, according to Reuters, trust in news is particularly low today [219], and previous research has highlighted the exposure to online disinformation in several political circumstances, from 2016 Constitutional Referendum to 2019 European Parliament elections [53, 87, 140, 141, 234]. A recent questionnaire by SOMA observatory on disinformation spreading on online social media[1] (a project

[1] Authors: Stefano Guarino, conceptualization, data collection, network analysis, writing; Francesco Pierri, conceptualization, network analysis, writing; Marco Di Giovanni, linguistic analysis, editing; Alessandro Celestini, data collection, network analysis, editing. [139]

[2] https://covid19.who.int

[1] http://www.t-6.it/report-on-the-role-of-the-information-in-the-emergency-covid-19-impacts-and-consequences-on-people-behaviors-report/

funded by the European Union) showed that people relied on official channels used by authoritative institutions in order to inform about the pandemic. Interestingly, social media were not the primary source of information during the crisis.

Similar to contemporary research, in this work we adopt a consolidated strategy to label news articles at the source level [39, 48, 137, 233, 238, 266] and investigate accordingly the diffusion of different kinds of information on Facebook. Thus, we use the term “disinformation" as a shorthand for unreliable information in several forms, all potentially harmful, including false news, click-bait, propaganda, conspiracy theories and unverified rumours. We use instead the term “mainstream" to indicate traditional news websites which convey reliable and accurate information. This approach has been mainly used for Twitter, which however exhibits a declining trend as a platform to consume online news [219, 234]. Similar to [129], we leverage Crowdtangle platform to collect posts related to COVID-19 from Facebook public pages and groups. We use a set of keywords related to the epidemic and we limit the search to posts in the Italian language. The overall dataset accounts for over 1.5 M public posts shared by almost 80k unique pages/groups. In particular, we are interested in understanding how specific disinformation narratives compete with official communications. To this aim, we further specify keywords related to three different controversial topics that have been trending in the past few months, all related to the origins of the novel coronavirus: (1) the alleged correlation between COVID-19 and migrants, (2) between the virus and 5G technology, and (3) rumours about the artificial origin of the virus.

The outline of this Section is the following: we first describe the methodology applied, including the collection of data from Facebook, the taxonomy of controversial topics, and text analysis tools (Section 6.1.2), then we describe our analysis (Section 6.1.3), and finally we draw conclusions (Section 6.1.4).

6.1.2 Methodology

Facebook data collection

We used CrowdTangle’s “historical data” interface [286] to fetch posts (in Italian language) shared by public pages and groups since January 1st 2020 until May 12th 2020 and containing *any* of the following keywords: *virus*, *coronavirus*, *covid*, *sars-cov-2*, *sars cov 2*, *pandemia*, *epidemia*, *pandemic*, *epidemic*. The tool only tracks public posts made by public accounts or groups. Besides, it does not track every public account[2] and does not track neither private profiles nor private groups. Our collection contains overall 1.59 M posts shared by 87,426 unique Facebook pages/groups. In the rest of the section, we use “accounts" as a shorthand to indicate the entire set of pages and groups.

[2] All pages with at least 100K likes are fully retained. For details on the coverage for pages with less likes we refer the reader to https://help.crowdtangle.com/en/articles/1140930-what-is-crowdtangle-tracking

Data is not publicly available, but it can be provided to academics and non-profit organizations upon request to the platform.

Controversial topics

In our analysis, we focus on three specific topics which were particularly exposed to disinformation during the infodemic[3]:

- **MIGRANTS**: conspiracy theories that attempt to correlate the spread of the virus with migration flows. These are mainly promoted by far-right communities to foster racial hate. Some of the related keywords are: *migranti*, *immigrati*, *ong*, *barconi*, *extracomunitari*, *africa*.

- **LABS**: rumours that have been used as political weapons to attribute the origins of the pandemic to the development of a bioweapon to be used by China and/or to undermine the forthcoming U.S. presidential elections. Some of the related keywords are: *laboratorio*, *ricerca*, *sperimentazione*.

- **5G**: hoaxes that can be summarized in two main streams, those claiming that 5G activates COVID-19 and those that deny the existence of the novel coronavirus and attribute its symptoms to reactions to 5G waves. Both lines are obviously false and not supported by scientific evidence. Some of the related keywords are: *5g*, *onde*, *radiazioni*, *elettromagnetismo*.

For the sake of simplicity, we will refer to an account as a "MIGRANTS" account (and likewise for the other topics) if the account shared at least $N = 2$ posts which contain a keyword in the related list (we selected $N = 2$ to reduce noise by discarding account that posted only once a tweet with that keyword). Finally, we denote any account as "controversial" if it is related to at least one of the three topics. In Table 6.1 we show a breakdown of the dataset in terms of posts and accounts. Note that the number of accounts is lower due to a pre-processing step described in the following paragraph.

Table 6.1: *Number of posts and accounts (groups and pages) for each controversial topic, and altogether.*

	5g	**Labs**	**Migrants**	**Intersection**	**Union**	**Total**
Posts	10937 (0.7%)	25695 (1.6%)	38486 (2.4%)	39 (0.024%)	72440 (4.6%)	1588536
Accounts	5493 (9.7%)	7076 (12.5%)	11238 (19.9%)	1958 (3.5%)	15865 (28.8%)	56436
Groups' Posts	5817	15278	21135	31	40175	715104
Groups	3194	4129	6571	1232	9007	28721
Pages' Posts	5120	10417	17351	8	873432	873432
Pages	2299	2947	4667	726	6858	27715

[3] https://www.newsguardtech.com/covid-19-myths/

Text analysis

We clean and pre-process posts' textual content as follows. Firstly, strings are lower cased and URLs, punctuation, emojis and Italian stop words (collected from *spacy* Python library) are removed. We also remove words related to the COVID-19 as they act as stop word for our analysis. Then, we tokenize texts using *nltk* Python library [28], and we remove tokens shorter than 4 or longer than 20 characters.

Then, we group tokens by account. To reduce noise effects we remove accounts with only 1 post and accounts with less than 20 tokens in total, obtaining 56,436 accounts from an original amount of 87,426. We compute the TF-IDF (Section 2.2.3) of the cleaned strings, neglecting tokens that appeared less than 5 times in the whole corpus. Finally, for each account we obtain a sparse 137,901-dimensional embedding vector.

6.1.3 Descriptive statistics

Posts statistics

We first inspect the prevalence of COVID-19 in online conversations by showing the time series of daily posts on Facebook in Figure 6.1. We observe a general increase in the overall volume (bottom), with a few spikes at the end of January (when China imposed lockdown), at the end of February (when the virus was first diagnosed in Italy), at mid March (when lockdown was applied in Italy) and at the beginning of May (when restrictions have been lifted). For what concerns controversial topics (top), we immediately see that volumes are negligible w.r.t general conversations (the same holds for daily interactions, which are 2 order of magnitude smaller); also, they are quite aligned in time but they do exhibit spikes of their own, which are most likely related to real world events (for instance sabotages of 5G antennas in several countries).

Polarization of accounts

To investigate the polarization [67] of accounts towards different topics we introduce a polarization score ρ defined as:

$$\rho = \frac{p_a - p_b}{p_a + p_b}$$

where p_b and p_a are, respectively, the number of controversial and non-controversial posts of the considered account. We define a *controversial post* any post that contains at least one of the manually selected tokens. The polarization index is constrained between -1, when all the posts of an account are about controversial topics ($p_a = 0$) and $+1$, when no posts involved controversial topics ($p_b = 0$).

Figure 6.2 shows the distribution of the polarization scores of accounts. We notice a trimodal distribution: a peak at $\rho = 1$ representing accounts not talking about controversial topics (the greater majority of accounts), a second peak at $\rho = 0$ that in-

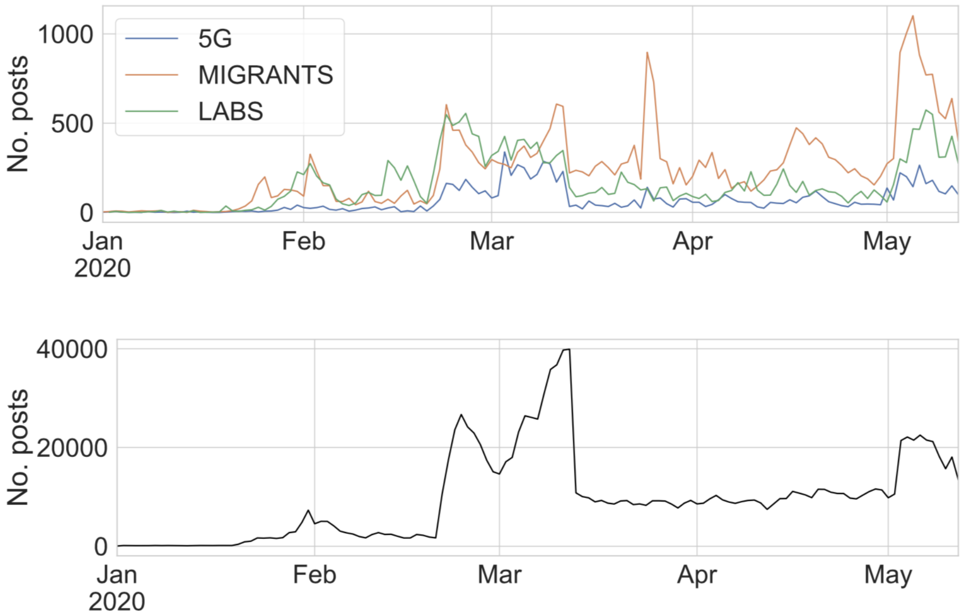


Figure 6.1: *Time series of the daily number of posts, total and per topic.*

cludes accounts talking equally about controversial and not controversial topics, and a third lower peak at $\rho = -1$ which represents accounts posting only about controversial topics.

In Figure 6.3 we also show how accounts are polarized when comparing topics against each other with the same rationale, *i.e.*, by defining p_a the number of posts about one controversial topic (*e.g.*, *5g*) and p_b the number of posts about a different controversial topic (*e.g.*, *Labs*). Peaks at $\rho = +1$ and $\rho = -1$ indicate that most accounts usually do not talk about more than one controversial topic.

Linguistic Analysis

In Figure 6.4 we show a kernel density plot for the embedding of accounts obtained as previously described in section 6.1.2. For visualization purposes, we select the first two PCA components. Red indicates controversial accounts, *i.e.*, accounts that published at least two posts about controversial topics, whereas blue indicates the remaining ones. Note that, even if the intersection of controversial accounts is negligible (see Table 6.1) and the accounts are usually polarized on a single controversial topic (see Figure 6.3), the embeddings of the two "classes" have similar distributions.

This result suggests that controversial themes are characterized by a common lexicon, distinct from the reminder of the dataset. The aforementioned embeddings might also be suitable input feature vectors for the definition of a finer classifier able to tell apart controversial posts and accounts not relying on predefined lists of keywords and/or news sources. The definition of a similar classifier is however beyond the scope

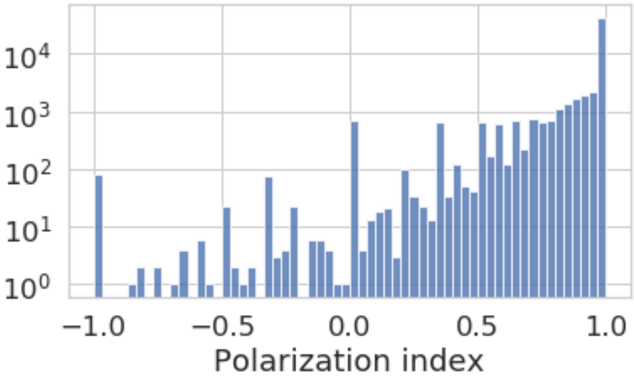


Figure 6.2: *Histogram of the polarization scores of accounts.*

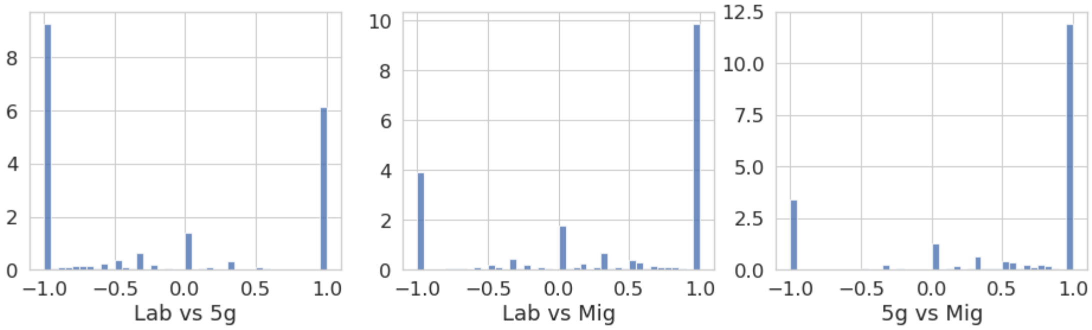


Figure 6.3: *Normalized histogram of polarization index of accounts, by couples of topics.*

of this paper and is left to future work.

Sentiment Analysis

We compute sentiments of posts using Neuraly's "Bert-italian-cased-sentiment" model[4] hosted by Huggingface [313]. It is a BERT base model [91] trained from an instance of "bert-base-italian-cased"[5] and fine-tuned on an Italian dataset of 45K tweets on a 3-classes sentiment analysis task (negative, neutral and positive) [20] obtaining 82% test accuracy. Previous work showed that text length can affect the classification accuracy of pre-trained models [8]. The model used in this analysis, however, performs extremely well also for texts of variable length and, albeit the model was trained using short texts (*i.e.*, tweets), it seems to benefit from the use of the entire available text (see Figure 6.5). To perform this analysis we used a Tripadvisor dataset[6] of 28754 Italian reviews of hotels and restaurants, with an average length of about 700 characters. As a consequence, the sentiment analysis is obtained truncating the texts at 1960 characters – a value identified experimentally as the optimal trade-off between efficiency and

[4] https://huggingface.co/neuraly/bert-base-italian-cased-sentiment

[5] https://huggingface.co/dbmdz/bert-base-italian-cased

[6] http://dbdmg.polito.it/wordpress/wp-content/uploads/2020/01/dataset_winter_2020.zip

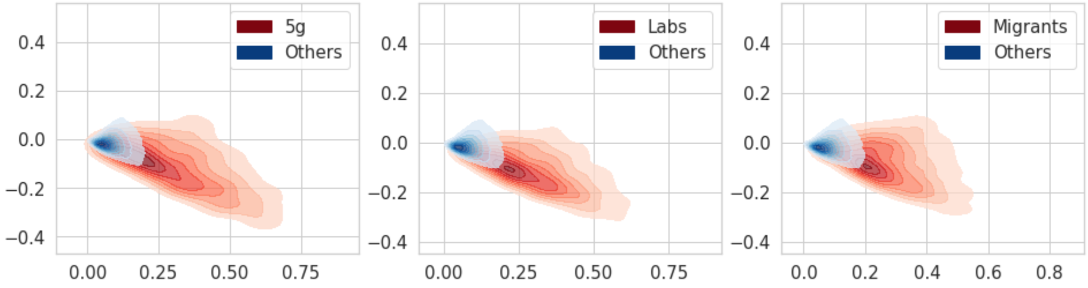


Figure 6.4: *Distribution of the first two main components of embeddings of accounts.*

accuracy, since using longer texts does not provide any measurable classification gain.

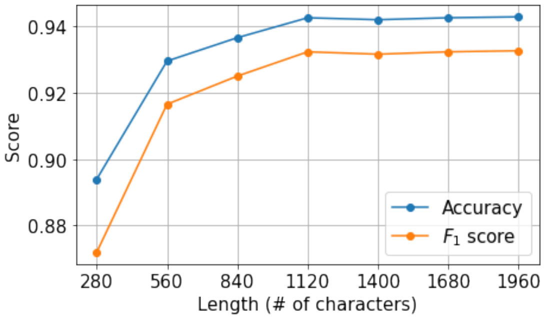


Figure 6.5: *Average accuracy and F_1 score of the sentiment classification model when texts are truncated at different lengths. The scores increase as we increase the truncation length, even if the resulting sentences are longer than the maximum length of sentences from our training set (280 characters).*

In Figure 6.6 we show how the general sentiment of posts evolves during the selected months by plotting the percentage of positive and negative posts weighted by the number of shares. We remark that, even if not shown in the figure, the great part of posts is classified as neutral (81.5%). This value decreases to 78.8% when the posts are weighted by their number of shares. Positive and negative peaks can be mapped to news and events, *e.g.* the two main peaks of negative sentiment, occurring on January 24 and February 10, match with the first confirmed COVID-19 cases in Europe and the first confirmed 1000 deaths worldwide, while the two main peaks of positive sentiment, occurring on February 2 and March 12, correspond to the successful isolation of the virus in the "L. Spallanzani" National Institute for Infectious Diseases and the diffusion of the #andràtuttobene ("it'll all work out") hashtag and slogan (see Figure 6.7).

6.1.4 Conclusions

In this chapter, I investigated online conversations about COVID-19 and related controversial topics on Facebook during a time window of 4 months. I analyzed more than 1.5 M posts shared by almost 80k groups and pages. After some general statis-

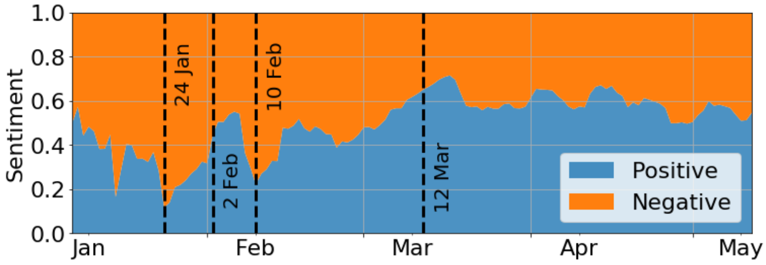


Figure 6.6: *7-days rolling average of the percentage of posts classified as positive or negative, weighted by the number of shares.*

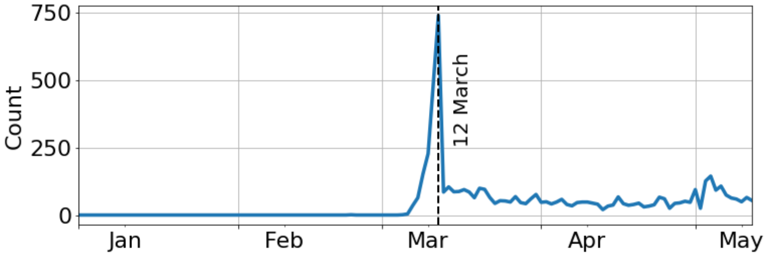


Figure 6.7: *Number of posts with the slogan "andràtuttobene" ("it'll all work out"). We notice a clear peak on March 12.*

tics of posts and users, I calculated how much users polarize between controversial and not controversial topics. I visualized their distribution and how the sentiment of posts changed over time. We obtained an explanatory description of our Facebook dataset that is useful to perform an in-depth analysis of the diffusion of disinformation during the COVID-19 outbreak in Italy.

In the following sections, we focus on a single aspect of the COVID-19 pandemic: vaccines. The first one describes how we monitor Twitter and Facebook and some preliminary investigations about misinformation, while the second one contains a work about the classification of pro-vax and no-vax Italian tweets about COVID-19 vaccines.

6.2 VaccinItaly: Monitoring Italian Conversations Around Vaccines On Twitter And Facebook[1]

6.2.1 Introduction

On January 30th, 2020, the World Health Organization declared the outbreak of a novel coronavirus (SARS-CoV-2) a global pandemic[7]. A year later, the spread of the virus has caused over 121 M confirmed cases and more than 2.5 M fatalities globally[8]. Italy, in particular, has been one of the first European countries to be hit by the virus, with over 3.28 M confirmed cases and 100 k fatalities as of March 2021, and the first country outside China to implement national lockdown to circumvent its spreading with severe social and economic consequences [35, 276]. Despite the global crisis, we witnessed the most rapid vaccine development for a pandemic in history when the Pfizer-Biontech vaccine showed a 95% efficacy and was approved in several countries[9] in late Fall, 2020. In the next few months, several other vaccines were going to be approved and made available to the public[10]. Italy, specifically, has started its vaccination campaign on December 27th, 2020, and reached over 6 M dispensed doses[11] as of March 13th, 2021.

As COVID-19 was spreading around the world, online social networks experienced a so-called "infodemic", i.e. an over-abundance of information about the ongoing pandemic, which yield severe repercussions on public health and safety [121,139,317,323]. It is believed that low-credibility information might drive vaccine hesitancy and make it hard to reach herd immunity [236, 317]. The European Social Observatory for Disinformation and Social Media Analysis has recently identified four macro-categories of unreliable information about COVID-19 vaccines[12]: (1) there haven't been enough tests on vaccines to guarantee their safety; (2) causal association for individuals who died after being vaccinated; (3) there are further medical complications due to vaccines; (4) vaccines can modify our DNA.

Since the 2016 US presidential elections, the research community has mostly focused its attention on political disinformation and election-related manipulation of online conversations [115, 183, 235, 266]. However, much concern has grown around

[1] Authors: Francesco Pierri: first author, conceptualization, implementation, experiment design, writing; Andrea Tocchetti: conceptualization, implementation; Lorenzo Corti: conceptualization, implementation; Marco Di Giovanni: conceptualization, implementation; Silvio Pavanetto: conceptualization, implementation; Marco Brambilla: conceptualization, editing; Stefano Ceri: conceptualization, editing. [239]

[7] https://www.who.int/publications/m/item/covid-19-public-health-emergency-of-international-concern-(pheic)-global-research-and-innovation-forum

[8] https://covid19.who.int

[9] https://www.pfizer.com/news/press-release/press-release-detail/pfizer-and-biontech-conclude-phase-3-study-covid-19-vaccine

[10] https://www.nytimes.com/interactive/2020/science/coronavirus-vaccine-tracker.html

[11] http://www.salute.gov.it/portale/nuovocoronavirus/dettaglioContenutiNuovoCoronavirus

[12] https://www.disinfobservatory.org/disinformation-about-covid-19-and-vaccines-a-journey-across-europe/

health-related misinformation which became manifest during recent measles outbreaks [116] and other epidemics such as H1N1 and Ebola [63, 119], eroding public trust in governments and institutions and undermining public countermeasures during such crises [84, 264].

In this paper, we describe VaccinItaly, a project to monitor Italian conversations around vaccines on multiple social media (Twitter, Facebook) with the aim of understanding the interplay between online public discourse and the vaccine roll-out campaign in Italy. Using a set of Italian vaccine-related keywords, regularly updated to capture trending hashtags and relevant events, as of March 13th, 2021 we collected over 3 M tweets and 1 M Facebook posts published by public pages and groups (we started our collection on December 20th, 2020). We provide public access to the list of keywords and tweet IDs[13], whereas access to Facebook data is granted by Crowdtangle [77] to academics and researchers upon request[14].

A specific goal of our project is to investigate the spread of reliable and unreliable information related to vaccines. Following a huge corpus of literature [39, 90, 137, 183, 317], we use a consolidated source-based approach to study how news articles, originated from low- and high-credibility websites, are shared alongside vaccine-related conversations on the two platforms. We also highlight YouTube as an additional potential source of misinformation about vaccines. Finally, we geolocate over 1 M users on Twitter and correlate their online activity with open data statistics about the Italian vaccine roll-out campaign[15].

Up-to-date results from our ongoing analyses are also available to the public through an online dashboard: http://genomic.elet.polimi.it/vaccinitaly/. A preview of the dashboard is available in Figure 6.8. This is similar in spirit to CoVaxxy[16] [90], a project based at the Observatory of Social Media (Indiana University) which aims to show the interplay between English-language online misinformation on Twitter and the US vaccine roll-out campaign. However, we focus on the Italian scenario and we also analyze Facebook data.

We believe that our project can contribute to a deeper understanding of the impact of online social networks in an unprecedented scenario where trust in science and governments will be critical to battle a global pandemic.

6.2.2 Related work

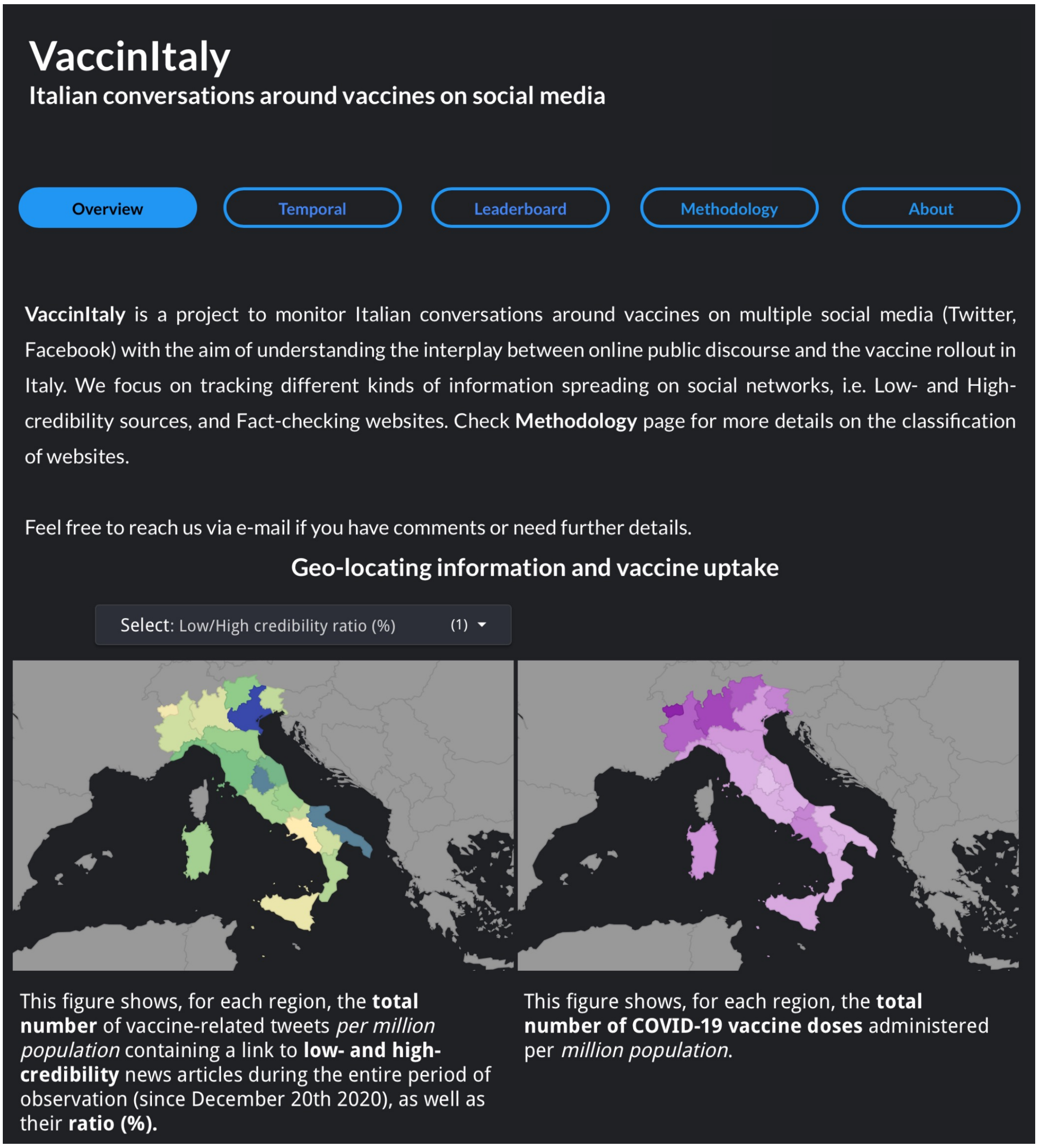
There is a huge interest, that is reflected into a large corpus of literature, around the diffusion of health-related (dis)information on online social networks. We describe a few contributions which are related to the Italian context and refer the reader to [306]

[13] https://github.com/frapierri/VaccinItaly

[14] Nevertheless, we provide a script to replicate our data collection using Crowdtangle keys.

[15] The data are available here: https://github.com/italia/covid19-opendata-vaccini

[16] https://osome.iu.edu/tools/covaxxy



VaccinItaly

Italian conversations around vaccines on social media

VaccinItaly is a project to monitor Italian conversations around vaccines on multiple social media (Twitter, Facebook) with the aim of understanding the interplay between online public discourse and the vaccine rollout in Italy. We focus on tracking different kinds of information spreading on social networks, i.e. Low- and High-credibility sources, and Fact-checking websites. Check **Methodology** page for more details on the classification of websites.

Feel free to reach us via e-mail if you have comments or need further details.

Geo-locating information and vaccine uptake

This figure shows, for each region, the **total number** of vaccine-related tweets *per million population* containing a link to **low- and high-credibility** news articles during the entire period of observation (since December 20th 2020), as well as their **ratio (%).**

This figure shows, for each region, the **total number of COVID-19 vaccine doses** administered per *million population.*

Figure 6.8: *Screenshot of the online dashboard associated to our project. Users can navigate through several sections, each providing different kind of analyses.*

for a deeper review of the existing literature on the subject.

The authors of [11] explored the relationships between Measles, mumps, and rubella (MMR) vaccination coverage in Italy and online search trends and social network activity from 2010 to 2015. Using a set of keywords related to the controversial link between MMR vaccines and autism, originated from a discredited 1998 paper, authors analyzed Google (search) Trends as well as the activity of Facebook pages and Twitter users on the same subject. They reported a significant negative correlation with the evolution of vaccination coverage in Italy (which decreased from 90% to 85% during the period of observation). They also identified real-world triggering events which most likely drove vaccine hesitancy, i.e. Court of Justice sentences that ruled in favor of a possible link between MMR vaccine and autism.

The authors of [100] provide a quantitative analysis of the Italian videos published on YouTube, from 2007 to 2017, about the link between vaccines and autism or other serious side effects in children. They showed that videos with a negative tone were more prevalent and got more views than those with a positive attitude. However, they did not inspect how videos were treating the link between vaccines and autism.

The authors of [254] analyze the Italian vaccine-related environment on Twitter in correspondence with the child vaccination mandatory law promulgated in 2017. Using a keyword-based data collection similar to ours, the author showed that the strong "politicization" of the debate was associated with an increase in the amount of problematic information, such as conspiracy theories, anti-vax narratives, and false news, shared by online users.

The authors of [75] also analyzed the debate about vaccinations in Italy on Twitter, following the mandatory law promulgated in 2017. They inspected the network of interactions between users, and they identified two main communities of people classified as "vaccine advocates" and "vaccine skeptics", in which they find evidence of echo chamber effects. Besides, they proposed a methodology to predicting the community in which a neutral user would fall, based on a content-based analysis of the tweets shared by users in the two groups.

6.2.3 Data collection

Twitter

Starting on December 20th, 2020, we use Twitter Filter[17] API to collect tweets matching the set of keywords in Table 6.2, in real-time. We routinely check for trending hashtags and relevant events to add new peculiar keywords, e.g. "#novaccinoainovax" and "#iononsonounacavia" were hashtags trending on specific days and consequently they were added to the list of keywords. The latter refers to vaccine advocates stating that no-vax should not be vaccinated, and the former indicates vaccine skeptics who "do not want to be guinea pigs for vaccines". The overall data up to March 13th, 2021 comprises approximately 3 M tweets shared by 258 k unique users.

Facebook

We used the *posts/search* endpoint of the CrowdTangle API [77] to collect public posts shared by pages and groups which matched the list of keywords previously defined,

[17] `https://developer.twitter.com/en/docs/twitter-api/v1/tweets/filter-realtime/api-reference/post-statuses-filter`

Table 6.2: *List of keywords used to filter relevant tweets and Facebook posts. They all refer to vaccines and vaccination in general, and some indicate specific pro and anti-vax views (e.g. "iononmivaccino" means "I will not get vaccinated", "vaccinareh24" means "Vaccinate all day long").*

vaccini	vaccinarsi	vaccinerai
vaccino	vaccinare	vaccineremo
vaccinazioni	vacciniamoci	vaccinerete
iononmivaccino	vaccinareh24	iononmivaccinero
vaccinazione	vaccinerò	novaccinoainovax
vaccinocovid	vaccinoanticovid	iononsonounacavia

resulting in over 10 M posts published by over 60 k public pages and groups, and reshared over 100 M times, as of March 13th, 2021. In the following, we will use the number of shares to compare Facebook with Twitter.

A limitation to our collection of Facebook is the coverage of pages and groups, whose data can be retrieved using the API. The tool includes over 6M Facebook pages and groups: all those with at least 100k followers/members and a very small subset of verified profiles that can be followed like public pages. Besides, some pages and groups with fewer followers and members can be included by CrowdTangle upon request from users. This might bias the data as, for instance, researchers and journalists might be interested in monitoring pages and groups sharing low-credibility thus leading to an over-representation of such content.

Sources of low- and high-credibility information

We extract URLs contained in tweets and Facebook posts to understand the prevalence of low- and high-credibility information shared in vaccine-based conversations [317]. We use a consolidated source-based approach to label news articles [90, 121, 137, 183, 233, 234, 237, 238, 266] depending on the reliability of the source, referring to two lists of Italian low- and high credibility news websites. The former corresponds to websites flagged by Italian fact-checkers for publishing false news, hoaxes and conspiracy theories[18]); the latter corresponds to Italian traditional and most popular news websites [301], and it is used as a reference to understand the prevalence of misleading and (potentially) harmful information. Lists are available in our repository[19], and we plan to manually augment them during our analyses.

We are aware that this approach, widely adopted in the research community, is not 100% accurate, as cases of misinformation on mainstream websites are not rare and, similarly, low-credibility websites do not publish solely "fake news". However, to date, it is the most reliable and scalable way to study misleading and harmful information.

[18] See www.pagellapolitica.it, www.facta.news and www.butac.it

[19] https://github.com/frapierri/VaccinItaly

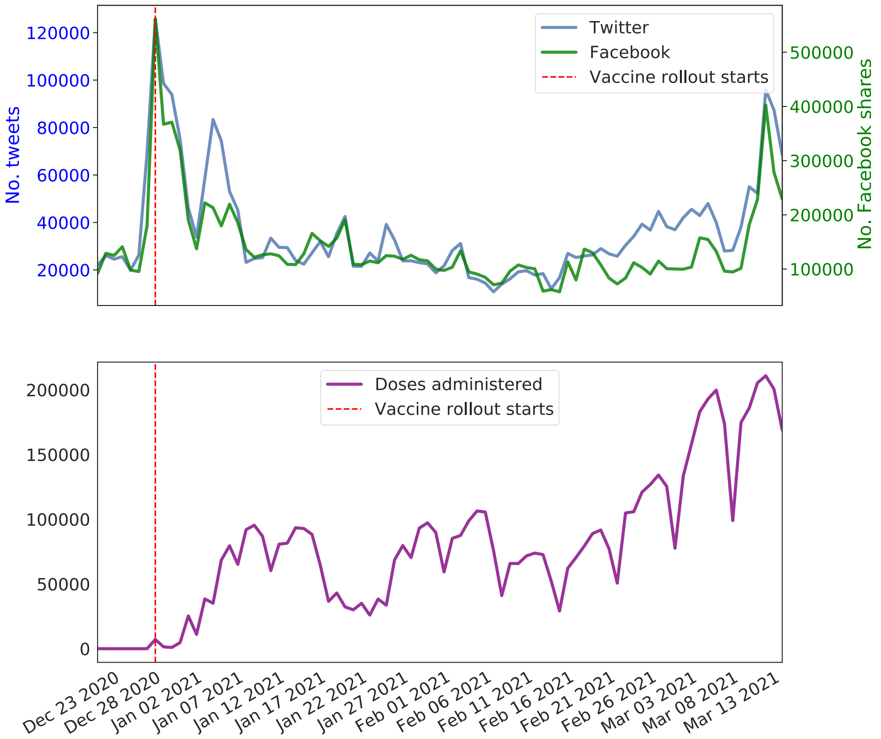


Figure 6.9: ***Top.*** *Temporal evolution of the daily volume of vaccines-related posts shared on both Twitter and Facebook. We use a dashed red line to indicate the beginning of the Italian vaccination campaign (27th December, 2020).* ***Bottom.*** *Total number of vaccine doses administered over time.*

Another limitation to our estimates is that our lists might not fully capture the amount of low- and high-credibility information circulating on Twitter. Besides, we do not consider different typologies of content such as photos, videos, memes, etc.

6.2.4 Online conversations and vaccine roll-out campaign

As previously mentioned, we started our collection on December 20th, 2020, in order to capture the beginning of the Italian vaccination campaign. A symbolic start took place on December 27th 2020, when a few thousand doses of Pfizer–BioNTech COVID-19 vaccine were used to vaccinate part of the medical and health personnel of hospitals, while a few days after 2021 New Year's eve over 300 k doses were delivered to Italy. In this ongoing phase, the priority is given first to health medical and administrative personnel, together with the guests and personnel of nursing homes, and then to elderly people and public service personnel.

Accordingly, we notice a huge spike in both Twitter and Facebook volumes following the symbolic start (over 120 k tweets and 500 k Facebook posts shared in a single day), and a slightly smaller spike after the actual beginning of the campaign (a peak of 80 k tweets and 400 k Facebook shares), as shown in Figure 6.9. As the number of doses administered increases to a steady level, we notice that public attention slowly

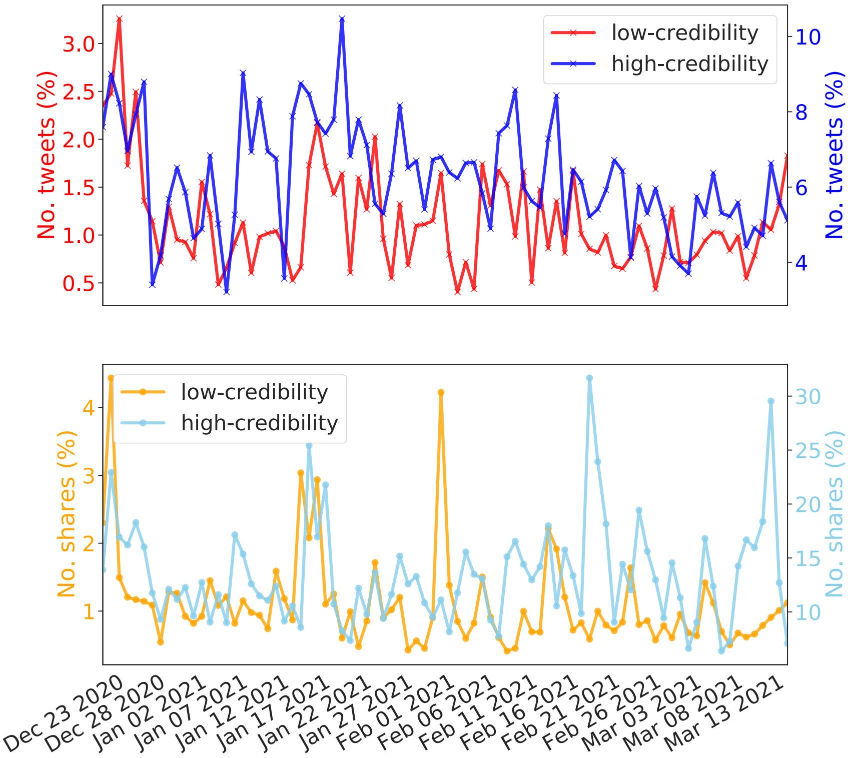


Figure 6.10: *Daily fraction of high-credibility and low-credibility content, compared to online conversations altogether, for Twitter (Top) and Facebook (Bottom).*

decreases. However, we notice a second surge of online conversations in March, in correspondence with the suspension of Astrazeneca vaccine in several European countries following an investigation of the European Medicines Agency about unusual blood disorders[20].

Overall we notice that the volume of vaccine-related conversations on Facebook is much higher than on Twitter, and this is probably due to the different size of their user base[21].

6.2.5 Prevalence of low- and high-credibility information

A key focus of our research is to analyze the spread of low-credibility information on social media, using high-credibility information as a reference. Overall, we report over 30 k tweets and 130 k Facebook shares linking to low-credibility news, and over 188 k tweets and 1.6 M Facebook shares linking to high-credibility news.

In Figure 6.10, we plot the fraction of tweets and Facebook posts shared that contain a link to either low- or high-credibility information. We note that the amount of low-credibility articles shared on both social media is much smaller compared to high-

[20] https://www.ema.europa.eu/en/news/covid-19-vaccine-astrazeneca-prac-preliminary-view-suggests-no-specific-issue-batch-used-austria

[21] https://www.statista.com/statistics/787390/main-social-networks-users-italy/

credibility, on both platforms. Relatively, the mean daily amount of low-credibility information is similar on the two platforms (1.13% on Twitter, 1.10% on Facebook), whereas, interestingly, the mean daily amount of high-credibility circulating on Facebook is higher compared to Twitter (6.27% on Twitter, 13.41% on Facebook). Given the limitations of our analysis, we might not simply state that the information spreading on Facebook is more reliable than on Twitter. Besides, the amount of low-credibility information is non-negligible on both platforms, and it might play a relevant role in shaping the public discourse and opinion around vaccines.

In Figure 6.11, we show a leaderboard of the top-20 news sources shared on Facebook and Twitter, considering both low- and high-credibility information. We also add the totality of low-credibility information. As previously noted, Facebook shares are an order of magnitude larger than Twitter. Besides, high-credibility domains are shared more than low-credibility websites on both platforms. Except for "liberoquotidiano.it", a right-wing news website which notably publishes misleading information, we notice the same two most shared low-credibility domains in the leaderboard, namely "imolaoggi.it" and "byoblu.it". The former is a well-known far-right-wing website that regularly publishes false news with nationalist and anti-immigration views, the latter is a blog that has been repetitiously flagged for sharing hoaxes about health-related subjects, including the COVID-19 pandemic.

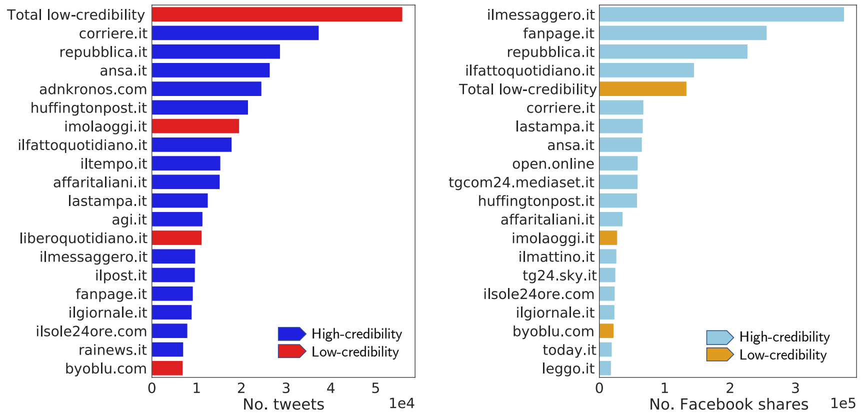


Figure 6.11: *Top-20 news w.r.t the overall number of Twitter (left) and Facebook (right) shares. We also indicate the total amount of low-credibility content and compare it with individual sources of high-credibility information.*

By looking at the overall amount of low-credibility news shared on both platforms, we notice interestingly that on Twitter this is larger than any individual high-credibility source. We do not observe the same for Facebook, where still total low-credibility is comparable to the top-3 high-credibility domains. This shows that even though trustworthy information is more prevalent on both social media, the amount of disinformation content shared is still remarkable.

Finally, we investigate the relative popularity of low-credibility news websites on the two platforms, by computing Spearman's correlation coefficient of the websites ranked by their volumes. We find a significant positive correlation for low-credibility websites (R= 0.65, p-value= $1.14e-05$), indicating that the majority of unreliable sources is popular on both platforms.

6.2.6 YouTube as a potential source of misinformation

As an additional source of information about vaccines, we consider links to YouTube videos shared alongside Facebook and Twitter posts. Previous work [100] has shown that YouTube is used by both vaccine advocates and skeptics, and we aim to investigate the quality of videos shared on the two platforms. Overall, our dataset contains over 6 k links to YouTube shared 21,407 times on Twitter and 132,553 on Facebook.

After extracting URLs pointing to YouTube from tweets and Facebook posts, we use their IDs to query the Youtube API and collect metadata available for such videos. We collected data for approximately 3 k videos (published by 1.6 k unique channels) shared on Twitter, and 3.2 k videos (published by 1.5 k unique channels) shared on Facebook. For approximately 300 videos (50 of which were present on both platforms) the API did not return any results, meaning these videos were removed from YouTube due to copyright or policy infringement. Such videos were shared over 800 times on Twitter and 6.5 k times on Facebook. Following [317], we argue that these videos might have contained suspicious and harmful content. However, we cannot confirm this hypothesis as they were deleted and are no longer available.

We manually inspected the top-20 videos based on the number of tweets and Facebook shares. On Twitter, these videos achieved a total of 5,511 retweets and 5,770,308 YouTube visualizations, while on Facebook they were shared 61,154 times and reached 11,521,158 visualizations. The number of YouTube views was extracted on March 18th. Interestingly, we find several popular videos, on both platforms, which are associated with anti-vax views and misleading information.

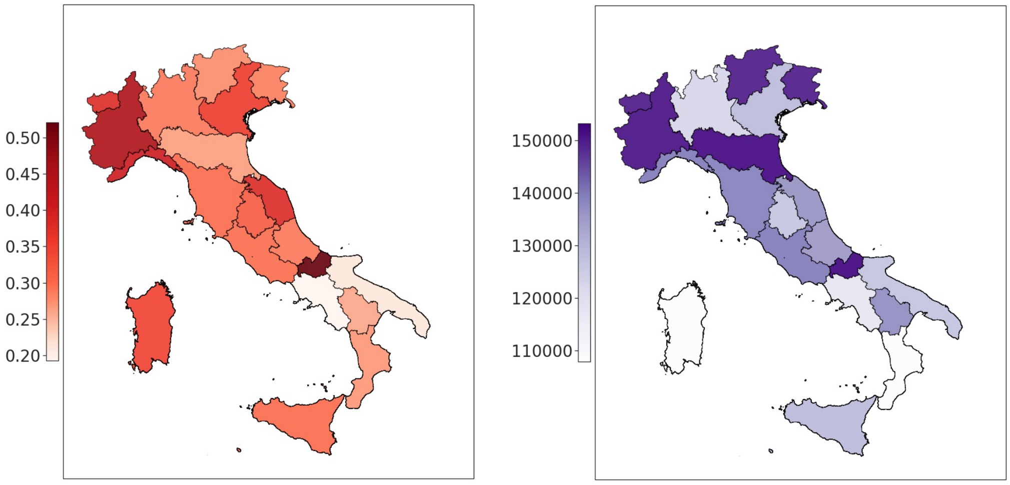
A relevant case is the 1st most shared video on Twitter (and 4th on Facebook) with the title "IL PARERE DEL PREMIO NOBEL LUC MONTAGNIER SULLA VACCINAZIONE ANTI-COVID [VIDEO IN ITALIANO]"[22], with over 700 retweets, 4k Facebook shares, and 450k YouTube views. In this video, the Nobel prize winner Luc Montagnier refers to Moderna company as "sorcerer apprentices" stating that they only tested the vaccine on animals, and it's thus not possible to foresee the effects of the vaccine on humans. He also proposes alternative natural treatments against COVID-19 and states that vaccinating the whole population is not the solution.

We omit other examples for reasons of space, but we report that several other popular

[22] Translation: "The opinion of Nobel Prize Winner Luc Montagnier on COVID-19 vaccination". Available at www.youtube.com/watch?v=kHGtn_vnrJ8.

videos mention conspiracy theories behind the origin of the virus and/or the effects of vaccines as well as proposing alternative therapies and suggesting the audience not to get vaccinated.

The fact that through a simple manual evaluation we encounter almost a dozen of suspicious and, in some cases, explicitly harmful videos among most popular videos indicates that further investigation is required. Indeed, it appears that YouTube is a potential source of online misinformation about vaccines.



Average % of low-credibility tweets.

Number of total doses per million population.

Figure 6.12: ***Left.*** *Average fraction of low-credibility tweets shared by users, for each Italian region.* ***Right.*** *Total amount of vaccine doses administered, per million population, in each Italian region.*

6.2.7 Geolocating Twitter conversations

A goal of our project is to link online conversations with geographical details on the ongoing vaccination campaign, e.g. the number of doses administered in each Italian region.

To this aim, we attempt to geolocate Twitter users by using a naive string matching algorithm, i.e. checking whether they have a "location" field disclosed in their profile and matching it against a list of Italian municipalities, provinces, and regions[23]. In the case of multiple matches, we retain the longest one. We matched about 16 k unique locations and, among over 135 k users putting a "location" in their profile, we accordingly geolocated 73 k users to either an Italian municipality or region. These shared over 1.3 M tweets. The number of accounts mapped to each Italian region is significantly positively correlated with the actual population (R$=0.89$, p-value< 0.001). However, it is known that the Twitter sample of users might not be fully representative of the Italian population, and this is a limitation to analyses that infer demographics from Twitter [4].

[23] Taken from the Italian National Institute of Statistics and available at `https://www.istat.it`.

As an illustrative example, we show in Figure 6.12 statistics on the amount of low-credibility information circulating on Twitter, and the status of the vaccination campaign. Specifically, in the left panel, we show the average fraction of low-credibility tweets shared by users geolocated in each region; darker colors correspond to higher values. We note that on average, Italian users share low-credibility information around 0.20-0.50% of the time. In the right panel, we show the total number of doses administered per million population, in each region. We can note that Lombardy is performing worse than most regions, even though it was the region most struck by the pandemic during the first wave.

These results are still preliminary, as the methodology presents several limitations and needs further assessment, e.g., how to handle multiple locations appearing in the "location" field of user profiles or when false places match with Italian municipalities with misleading names (e.g. "Paese" which translates as "village").

6.2.8 Conclusions

We present an ongoing project which monitors online conversations of Italian users around vaccines on Twitter and Facebook. We give full access to the data we collect, and we provide up-to-date results in online interactive dashboards. Preliminary analyses show a non-negligible amount of low-credibility information circulates on both platforms, and they indicate YouTube as a potential source of misinformation about vaccines. Our final goal is to understand the interplay between the public discourse on online social media and the vaccine roll-out campaign. In particular, we aim to investigate the impact of online sentiment (e.g. communities of pro and anti-vax) and misinformation about vaccines on vaccine uptake in Italy. We also aim to check whether there are geographical and socio-economical differences shaping online conversations and vaccination campaigns.

6.3 A Content-based Approach for the Analysis and Classification of Vaccine-related Stances on Twitter: the Italian Scenario[1]

6.3.1 Introduction and Related Work

A year after the outbreak in China, the SARS-CoV-2 has radically changed our lives, and despite the countermeasures adopted by countries across the world to prevent its spreading [35,276], the pandemic has infected more than $123M$ individuals and caused more than $2.7M$ deaths worldwide[24]. Nevertheless, we have seen the rapid development of several vaccines with over 90% effectiveness, the foremost being the one developed by Pfizer-BioNTech, announced in November 2020[25]. As of March 22nd, 2021, more than $439M$ vaccine doses have been administered worldwide, which translates to almost 5.7 doses every 100 individuals[26]. Italy, in particular, has started its vaccination program on December 27th 2020, with $8M$ doses given to citizens[27] as of March 22nd, 2021.

Although vaccination is worldwide considered one of the greatest achievements of public health, it is still perceived as unsafe and unnecessary by a growing number of individuals and the causes of this phenomenon involve emotional, cultural, social, spiritual, political and cognitive factors [104]. In particular, after the decline in measles coverage in 12 European countries in 2018, vaccine hesitancy has been included in the top-10 threats to global health in 2019 by the World Health Organization [28].

Over the last decades, social media experienced a quick growth in their user-base and daily usage. Echo chamber effects, i.e. reinforcement of users' beliefs via the interaction with a closed set of similar users, have been observed during debates about political and socially relevant topics [70, 88]. The authors of [75] observed a similar phenomenon regarding Italian Twitter conversations about vaccines in 2019, focusing on the worrying asymmetry of the chambers' topology.

The alarming growth of skepticism, powered by social media, caused an increase of scientific contributions inspecting the phenomena from different points of view. The authors of [236] deeply studied online misinformation about vaccines in US while the authors of [167] constructed semantic networks of vaccine information from highly shared websites of Twitter users in the United States and the authors of [81] trained an SVM classifier to detect the stance of tweets about vaccines. Asymmetric behaviour

[1] Authors: Marco Di Giovanni: first author, conceptualization, implementation, experiment design, writing; Lorenzo Corti: conceptualization, implementation, writing; Silvio Pavanetto: conceptualization, writing; Francesco Pierri: conceptualization, implementation; Andrea Tocchetti: conceptualization, writing; Marco Brambilla: conceptualization, editing. [97]

[24] https://gisanddata.maps.arcgis.com/apps/opsdashboard/index.html#/bda7594740fd40299423467b48e9ecf6

[25] https://www.pfizer.com/news/press-release/press-release-detail/pfizer-and-biontech-conclude-phase-3-study-covid-19-vaccine

[26] https://www.nytimes.com/interactive/2021/world/covid-vaccinations-tracker.html

[27] https://www.governo.it/it/cscovid19/report-vaccini/

[28] https://www.who.int/news-room/spotlight/ten-threats-to-global-health-in-2019

of defenders and critics of vaccines in the French-speaking Twitter was discovered in [123]. Finally, the effect of bots and trolls in the debate is studied in [50]. Specific to the Italian context, many contributions have been published after the Law on Mandatory Vaccinations in 2017 [100, 195, 254].

In this work we inspect the SARS-CoV-2 vaccination debate on Italian-speaking Twitter from a textual content point of view. Our goal is to train an accurate stance classifier that detects patterns in tweets shared by supporters and skeptics of the vaccine. We design a semi-automated, human-in-the-loop, hashtag-based approach to label a large set of Italian tweets (similar to the one described in Section 5.2). We inspect the obtained labeled dataset by focusing on the location and date of tweets, and lexical patterns, looking at possible correlations and induced biases. Finally, we successfully train a BERT [91] model to classify the stance of tweets (“No-Vax" vs “Pro-Vax"), observing high values of AUROC and F1 score also on a dataset of manually labeled tweets that cannot be classified by the semi-automated approach previously defined. Our model can be used to monitor on real time the vaccination debate, independently on both the shared trending hashtags and the underneath social graph.

6.3.2 Data Collection

We collected data from **Twitter**, a micro-blogging platform widely used in Italy to discuss trending topics, whose official API allows for a fast implementation and a comprehensive collection. We query Twitter’s Streaming API searching for Italian tweets containing at least one of the keywords reported in Table 6.2 (see Section 6.2). The collection is running continuously since December 20th 2020, and by March 12th 2021 we obtained about $3M$ tweets shared by $250k$ different users [239].

6.3.3 Data Labeling

The manual labeling of big datasets is an expensive and non-scalable approach. Graph-based approaches have obtained impressive results when applied to detect stances in controversial debates [75, 84, 125]. However, these approaches are mainly used to categorize users, scoring their membership with respect to one side of the debate, but not to label *single* tweets.

We design a content-based, human-in-the-loop approach to semi-automatically label large sets of tweets as “Pro-Vax" or “No-Vax". This approach is based on *hashtags*, often used to express the stance of users about a topic [209]. Trending hashtags attract audience and get the attention of other users in the social network[29].

We define as **Gold hashtags** those that clearly indicate either a positive or negative view in the vaccine debate. We collect two sets of Gold hashtags, one for each side of

[29] Twitter has a specific section for trending hashtags and keywords `https://twitter.com/explore/tabs/trending`

Table 6.3: *Translated examples of tweets containing both a starting Gold hashtag (#iomivaccino or #iononmivaccino) and #novax (A and B) or #novaccinoainovax (C and D). Note that the two hashtags cannot be selected as Gold hashtags since they are used with different purposes by users from both sides of the debate.*

	#NoVax
A	In a world of #conspiracytheorists and **#novax**, me and my brother-in-law bet on who between the two of us would get vaccinated first. If there won't be hitches, this afternoon I will win the net. **#Iwillgetvaccinated** #vaccine
B	Please get vaccinated..so in the Movie 2022.... "the survivors " You won't be there **#Iwillnotgetvaccinated #novax**
	#NoVaccinoAiNoVax
C	**#Iwillgetvaccinated** even 17 times, to save the world from the terrible pandemics #COVID-19. **#NoVaccineToNoVax**, they don't deserve the help of science, rather #donateVaccineToAMigrant let's help them and be inclusive
D	Still relevant, I share to wake up some sleeper **#Iwillnotgetvaccinated** #IamNotAGiuneaPig **#NoVaccineToNoVax**

the discussion and we label tweets according to the hashtags they contain. We set the stance of a tweet based on the stance of the Gold hashtag, whereas tweets containing at least one Gold hashtag from both sets are discarded. To obtain the final set of Gold hashtags, we start from two Gold hashtags, one for each stance: #iomivaccino ("I will get vaccinated") and #iononmivaccino ("I will not get vaccinated"). Three annotators manually validate this selection by inspecting 50 tweets for each hashtag, finding only 2 tweets that clearly belong to the opposite stance.

We iteratively add new hashtags by searching from the most frequent co-occurring ones, manually selecting the most pertinent ones and choosing them based on their meaning. An example of discarded hashtag is #vanityfair (name of a fashion magazine), highly co-occuring with #iomivaccino, since we cannot safely assume that it is used only by supporters. We also discard hashtags that equally co-occur with hashtags from both sides in similar percentages. An example is #novax, that co-occurs with both #iomivaccino and #iononmivaccino about 50 times in original tweets (tweets that are not retweets). By manually inspecting tweets which shared this hashtag, we notice that it is used by skeptical users to state their side, but also by supporters to refer to their opponents (e.g., Table 6.3 A-B).

Finally, three annotators manually validated the selected hashtags, as previously described for the initial Gold hashtags. A hashtag is not validated (and thus discarded) if any annotator classified more than 10% of the associated tweets as belonging to the opposite class. In this way we reliably discard hashtags that are meant to be used by a specific side of the debate, but are also often used by the other side in a criticizing or ironic manner. An example is #NoVaccinoAiNovax ("No Vaccine To No-Vax"), that is used by “Pro-Vax" partisans in an attempt to prevent people, currently against vaccines, to change their minds and get vaccinated in the future. However, it is also largely used by “No-Vax" users to remark that they do not want to get vaccinated (e.g., Table 6.3 C-D). After three iterations we obtain a final set of three “Pro-Vax" Gold hashtags and three “No-Vax" Gold hashtags, shared in almost $50k$ original tweets, by manually labeling only a few hundreds tweets (statistics of the Gold hashtags are reported in Table 6.4). Since no other hashtag among the 50 most-frequent ones passes the validation procedure, we end the labeling process.

Table 6.4: *Statistics related to Gold hashtags: N is the total number of collected tweets, p_{orig} is the percentage of original tweets (tweets that are not retweets), N_u is the number of unique users that shared the hashtag, p_u is the percentage of unique users that shared the hashtag in an original tweet. Translations from top to bottom: IWillGetVaccinated, VaccinateH24, LetsNetwork; IWontGetVaccinated, IamNotAGuineaPig, HealthDictatorship.*

Gold Hashtag	N	p_{orig}	N_u	p_u
#iomivaccino	2810	0.71	1185	0.62
#vaccinareh24	29936	0.4	8828	0.46
#facciamorete	5896	0.35	1652	0.16
Tot Pro-Vax	37682	0.38	11269	0.43
#iononmivaccino	4231	0.39	1201	0.25
#iononsonounacavia	752	0.54	183	0.26
#dittaturasanitaria	6348	0.39	1388	0.3
Tot No-Vax	10886	0.31	2651	0.26

6.3.4 Data Description

In this section we investigate the geographical, temporal, and lexical distribution of our labeled tweets, looking for relationships and correlations with the computed stance.

Geographical Analysis

What is the geographical distribution of users who tweet about pro- or anti-vax views?

Twitter offers to its users the possibility to geographically tag shared tweets, but many users do not usually enable this functionality. For example, in our dataset only 881 tweets are geolocalized (0.03% of the total data). To investigate the geographical provenance of our data, we devised an approach to obtain the Italian region in which a tweet was posted, by looking at the location of users as specified in their profiles. We use a basic string matching algorithm to match it with the names of the 20 Italian regions, the 107 provinces and the 7903 municipalities[30], also including the most common English translations (e.g., Milan, Tuscany). We obtained the locations $1.6M$ of

[30] https://it.wikipedia.org/wiki/Comuni_d%27Italia

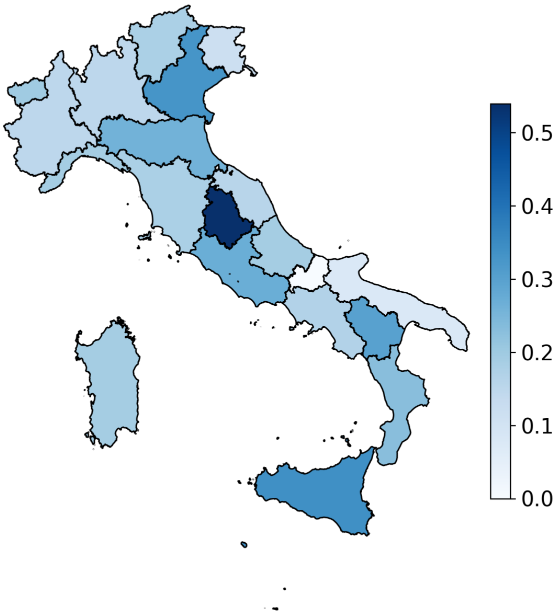


Figure 6.13: *“No-Vax" vs “Pro-Vax" ratio of geolocated tweets. The darker the color, the higher the fraction of “No-Vax" tweets shared from that region.*

tweets (including retweets), $19k$ of which also contain one gold hashtag. Figure 6.13 shows the geographical distribution of the ratio between the number of tweets using a “No-Vax" gold hashtags and the number of tweets using a “Pro-Vax" gold hashtags. We note that Umbria is the region with the highest “No-Vax to Pro-Vax" ratio, with only about twice as much “Pro-Vax" tweets compared to “No-Vax" ones.

Temporal Analysis

What is the temporal dynamics of the two factions?

The data analysed in this study spans the months from 20/12/2020 until 12/03/2021. Figure 6.14 shows the daily ratios of tweets labelled as “No-Vax" versus the ones labelled as “Pro-Vax", using as reference the gold hashtags from Table 6.4. We notice a steep valley at the beginning of January, since #vaccinareh24 was trending, and a spike at the beginning of February, most likely due to a debate about vaccines between Dr. Amici and Dr. Bassetti, broadcasted live during an episode of *Non è l'arena* on La7 (an Italian mainstream television channel). The conflict between doctors fueled the controversy and resulted in an influx tweets with a skeptical inclination about the vaccine.

Lexicon Analysis

What is the lexicon overlap between tweets shared from the two factions?

Our goal is to train an accurate stance classifier of tweets. A big lexicon overlap between training texts belonging to opposite classes forces a classifier to learn the

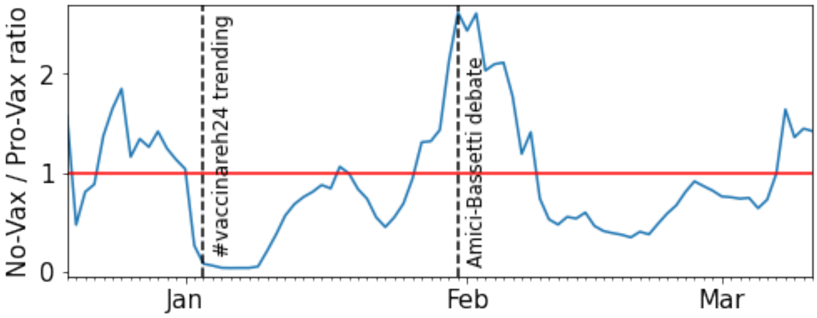


Figure 6.14: *Daily ratio between 7-day Moving Averages of "No-Vax" occurrences and "Pro-Vax" occurrences. The red line indicates the same amount of "No-Vax" and "Pro-Vax" shared tweets.*

Table 6.5: *Translated top-5 words ranked by PMI (Pointwise Mutual Information) scores and the proportion of texts in each class containing each word.*

No-Vax	%	Pro-Vax	%
fakepandemic	1.7	wespreadinformation	2.1
everybodyaccomplice	1.6	fuckcovid	1.5
firstdonotharm	1.4	4january	1.3
vaccinationpassport	0.7	reportvaccines	1.1
whensciencekills	0.6	vaccinesanticovid19	1.1

meaning of sentences. On the other hand, if the lexicon overlap is small, a classifier could rely on the presence of specific, often unrelated, words to make the right prediction. We quantify the lexicon overlap of the two classes by computing the Pointwise Mutual Information (PMI) between words and classes [146]: a word has a high PMI score with respect to a class when that word occurs mainly in tweets from a single class (e.g., a word used only by "No-Vax" users). For this analysis, we discard Italian stop words and apply text tokenization using the NLTK library [28]. We report in Table 6.5 the first five tokens for each class, ranked by PMI score, and the proportion of texts in each class containing each token. In both datasets, the frequency of tokens with large values of PMI is low, meaning that tweets belonging to the two classes use mostly similar lexicons. The most frequent token found among the ones with high PMI score is "facciamoinformazione" ("we spread information"), that is found only in 2.1% of the texts labeled as "Pro-Vax". Therefore, a classifier cannot safely rely on the presence of specific words since the most indicative ones are not very frequent.

6.3.5 Stance classification

Data Cleaning

Before training the classifier, we cleaned the text of tweets through the following procedure. Texts are lowercased, URLs are removed and spaces are standardized. **We**

remove Gold hashtags (Table 6.4) since they were used to automatically label tweets, thus maintaining them will introduce a strong bias in the trained models. Tweets containing at least half of the characters as hashtags are also removed, since they are too noisy. To prevent overfitting we remove duplicate texts, including retweets. We also remove texts shorter than 20 characters, that usually refer to URLs or other tweets, being difficult to understand and contextualize. The cleaning procedure reduces the number of tweets to about $10k$, of which $1.8k$ labeled as “No-Vax".

Methodology

Given the large set of labeled tweets using Gold hashtags, we train six text classifiers to predict the stance of a tweet. We select the following models:

- Majority classifier (Baseline);
- Logistic regression and SVM, both fed with TF-IDF of Bag of Words vectors [111, 163] (for further details see Section 2.2.3);
- FastText [165], a fast baseline approach widely used for text classification. Its architecture is similar to the CBOW model in Word2Vec [203]. It is known to reach performances on par with some deep learning methods, while being much faster (for further details see Section 2.3.2);
- BERT [91], a Transformer-based model [298] that reaches state-of-the-art performances on many heterogeneous benchmark tasks. The model is pre-trained on large corpora of unsupervised text using two self-supervised techniques: Masked Language Models (MLM) task and Next Sentence Prediction (NSP) task. Pre-trained weights are available on the Huggingface models repository [313](for further details see Section 2.4.4). We select a model pre-trained on a concatenation of Italian Wikipedia texts, OPUS corpora [288] and OSCAR corpus [221], performed by MDZ Digital Library[31]. We fine-tune the model on our data[32]. This pre-trained model has knowledge of the Italian language, lexicon and grammar, but it has few information about our topic (SARS-CoV2 vaccine). We apply Adaptive fine-tuning (AF) [258] to tackle this issue. The pre-trained Italian BERT is unsupervisedly fine-tuned on a MLM task using our full dataset (removing retweets to prevent overfitting). We obtain a *specialized* model about COVID-19 vaccine, that is fine-tuned on supervised data (like the original Italian model). We refer to this configuration as BERT+AF.

[31] https://huggingface.co/dbmdz/bert-base-italian-xxl-uncased

[32] Fine-tuning performed on a single NVIDIA Tesla P100, for 10 epochs. Best weights selected by minimizing the evaluation loss. Learning rate (10^{-5}) set through grid search.

Results

In Table 6.6 (left) we report Area Under ROC, weighted F1 score and F1 score on the “No-Vax" class. The values are average of 5-fold cross validation on the training set obtained with Gold hashtags. As expected, the BERT+AF model obtains the best results.

To test the generalization capabilities of our classifiers, we feed them with a Test set of 1000 *general* tweets: tweets that does not contain any Gold hashtags. The tweets are manually labeled by three annotators in four classes: “Pro-Vax", “No-Vax", “Neutral" and “Out of Context". We removed “Neutral" and “Out of Context" tweets obtaining 412 tweets, of which 132 labeled as “No-Vax". In Table 6.6 (right) the metrics confirm that BERT+AF handles general tweets better than the baseline models, thus can be applied to detect and prevent the spread of negative and harmful messages.

Table 6.6: *Validation and Test performance of classifiers.*

	Validation			Test		
Model	AUROC	$F1_w$	$F1_{novax}$	AUROC	$F1_w$	$F1_{novax}$
Baseline	0.50	0.74	0	0.50	0.55	0
LR	0.83	0.83	0.39	0.71	0.67	0.36
SVM	0.83	0.84	0.45	0.73	0.71	0.47
FastText	0.75	0.81	0.32	0.71	0.60	0.16
BERT	0.89	0.87	0.60	0.76	0.73	0.54
BERT+AF	**0.93**	**0.89**	**0.68**	**0.80**	**0.75**	**0.60**

6.3.6 Conclusions

The first step to fight the spread of misinformation is the detection of harmful messages. In this work, we collected and analyzed tweets about the Italian vaccination campaign. We designed a hashtag-based approach to semi-automatically label tweets, and we analyzed the labelled data in terms of geographical, temporal, and lexical distribution. Finally, we used them to train a BERT-based binary-classifier. Our approach suffers from some limitations. First, the selection and usage of Gold hashtags have a strong relationship with the date they were trending. Results on the test set suggest small overfitting, but further investigations are required to confirm its relevance. Second, by construction, the training dataset, our classifier does not detect neutral tweets or tweets whose stance is undefined, even if widely shared. In future works, I will implement a 3-class classifier, including the Neutral class, and the real-time application of the obtained model to promptly detect the daily trend of “No-Vax" tweets.

CHAPTER 7

Conclusions

This thesis collects the main works that I have performed during my PhD. I decided to investigate human-generated textual data collected from public social networking sites, using both classic and recent NLP techniques. The main goal of these works is to find the best representation of users and posts. I select, train and use Language Models to obtain informative embeddings used later in higher-level tasks. The heterogeneity of tested tasks suggests that these techniques are powerful and flexible, easily adaptable to different situations. However, often state-of-the-art models trained on formal datasets underperform when applied to noisy and unsupervised datasets from social networks. The selection of the appropriate approaches and how to train them is not straightforward.

I started my investigation researching how to extract emerging knowledge from social networks, observing that classical approaches such as Term Frequency vectorization, applied to selected and cleaned inputs (i.e., Proper Nouns), generate informative representations of users. The designed pipeline successfully finds Twitter accounts similar to those selected as seeds by experts, thus automatically extracting previously unknown information about different domains from the social network. The iterative implementation of the pipeline allows a continuous extraction that geographically spans all over the world, including always novel and emerging information. The knowledge extracted could automatically update knowledge bases.

The computation of distances and similarities between users is a crucial step of the pipeline. Nowadays, many alternative approaches are available to define similar users, so I investigated the best methodologies. In this thesis, I focus on similarities between content generated and shared by users, neglecting other information, such as the underneath social graph, demographic features and temporal quantities. The first research focuses on BoW approaches applied to both syntactic and semantic features. It results in performances highly topic dependent when applied to community detection and characterization tasks. Different communities of users behave differently on social networks, so they are easier or harder to detect and categorize.

When communities of users write about the same topic in opposite ways, we talk about controversies. I investigated the detection and quantification of controversial topics on social media designing a content-based pipeline that outperforms graph-based ones. I hypothesize that, even if the underlying graph reflects users' opinions about a topic, the textual content shared by the users should too. I selected 30 multilingual trending topics and applied the pipeline quantifying the controversy score of each one of them. This task is challenging because the straightforward application of recent deep Language Models such as BERT does not outperform classical and faster approaches. The selection of the best Language Model to embed users and compute similarities is crucial here. However, both techniques improve the graph-based baselines without sacrificing language independence.

However, deep Language Models obtain state-of-the-art results in a large variety of tasks when suitably trained. Thus I decided to investigate the best approaches to obtain more meaningful embeddings of data from social networks with Transformer-based models. The first research focuses on single posts exploiting Twitter's intrinsic powerful signals of relatedness: replies and quotes of tweets. The trained model reaches state-of-the-art performances on Semantic Textual Similarity tasks on noisy data from social media and accuracies comparable to previous models on formal data. This model allows the generation of accurate representations of single posts, but the final goal is the design of a pipeline to embed whole users. I proposed a Hierarchical Approach, whose Stage-1 model is the one just obtained, while the Stage-2 model is a small Transformer-based model that outperforms straightforward alternatives. The second main contribution of the work is the design of a large corpus used as an evaluation set without the need for human annotators, making the whole process completely reproducible.

Finally, I describe two applications of content-based approaches involving online discussions about political events and COVID-19 misinformation and vaccines. The first application involves the classification of the political stance of users and posts. The first work involves the classification of the political inclination of Italian deputies. I applied classical embedding techniques so to obtain insights from the trained models. The second work focuses on the 2020 Italian constitutional referendum. I analyzed the online discussion by collecting a complete dataset about the topic and investigating its possible biases. I trained an accurate classifier using a ground-truth obtained through hashtag-based semi-automatic labelling, and I finally discussed the discrepancy between the online activity and the referendum outcome.

The second application involves online discussions about COVID-19. Due to the recent COVID-19 emergency, I investigated the related infodemic problem in Italy since misinformation circulates through Facebook. I analyzed the public textual content, computing polarization, distribution and sentiment of the users and posts shared during the first four months of the emergency. The main goal of the work is to understand how information and misinformation spread to prevent and react against the “infodemic". It was performed jointly with other researchers investigating the spread of URLs coming from known misinformation sources and the structure and proprieties of the underneath graph structure, not reported in this thesis. This study shows that users sharing misinformation are similar, independently of the kind of misinformation involved. This work led to a more significant project analyzing the interplay between online public discourse and the vaccine roll-out campaign in Italy: Vaccinitaly. My role in the project concerns the collection and investigation of the textual content of tweets. I started by designing a Transformer-based stance classifier to detect Pro-vax and No-vax tweets, exploiting adaptive finetuning to obtain accurate results. In future works, we will also develop a three-classes classifier including neutral posts, and we will extend the analysis to other European countries with a multilingual approach.

Due to the successful application of language models to human-generated textual data, **future works** will focus on three main directions.

First, refined embeddings of tweets and users are always necessary. As techniques evolve, the design of better embedding approaches is always possible. The simple substitution of novel and better language models on previous pipelines is not always enough, and specific training procedures on appropriate datasets are essential. Unsupervised or self-supervised approaches are the most promising techniques to investigate since manually labelling large volumes of data is usually expensive and inefficient. The inclusion of features from the social graph proved to be effective in automatically supervising the pre-training stage. Finally, including features other than the textual content should improve the results, but the similarity between users in this setting is hard to define.

Second, I will continue to consider novel applications of similarities between tweets and users. Interesting perspectives come from both already existing tasks, such as link prediction of users based on how similar they write, or novel tasks, preferably inspired by contemporary events such as the political elections and current debates about COVID-19 vaccine-related Green Pass.

Third, investigating and quantifying the noise of texts is particularly important when data from social media are analyzed. Even if it is not easy to train robust models, text augmentation [92, 93] is a promising technique to control the amount and typology of noise in the training datasets. Moreover, the influence of emojis, typos and slang should be prioritized so that Language Models can better adapt to human-generated data.

Concluding, the future is promising. The field is continuously evolving while the research community is active. Everyday novel works, models, training techniques, tasks and datasets are publicly released, available to researchers and companies. Our knowledge of Natural Language and our ability to use that knowledge to teach machines proper ways to process that kind of information is exponentially increasing. The help of social networking sites that allows people to generate enormous magnitudes of data is essential to keep this trend growing.

I cannot wait to see what we will learn and build in the future, hoping that my contribution will be significant and my ideas will be understood, accepted and used by the rest of the community.

APPENDIX A

Auxiliary Tables

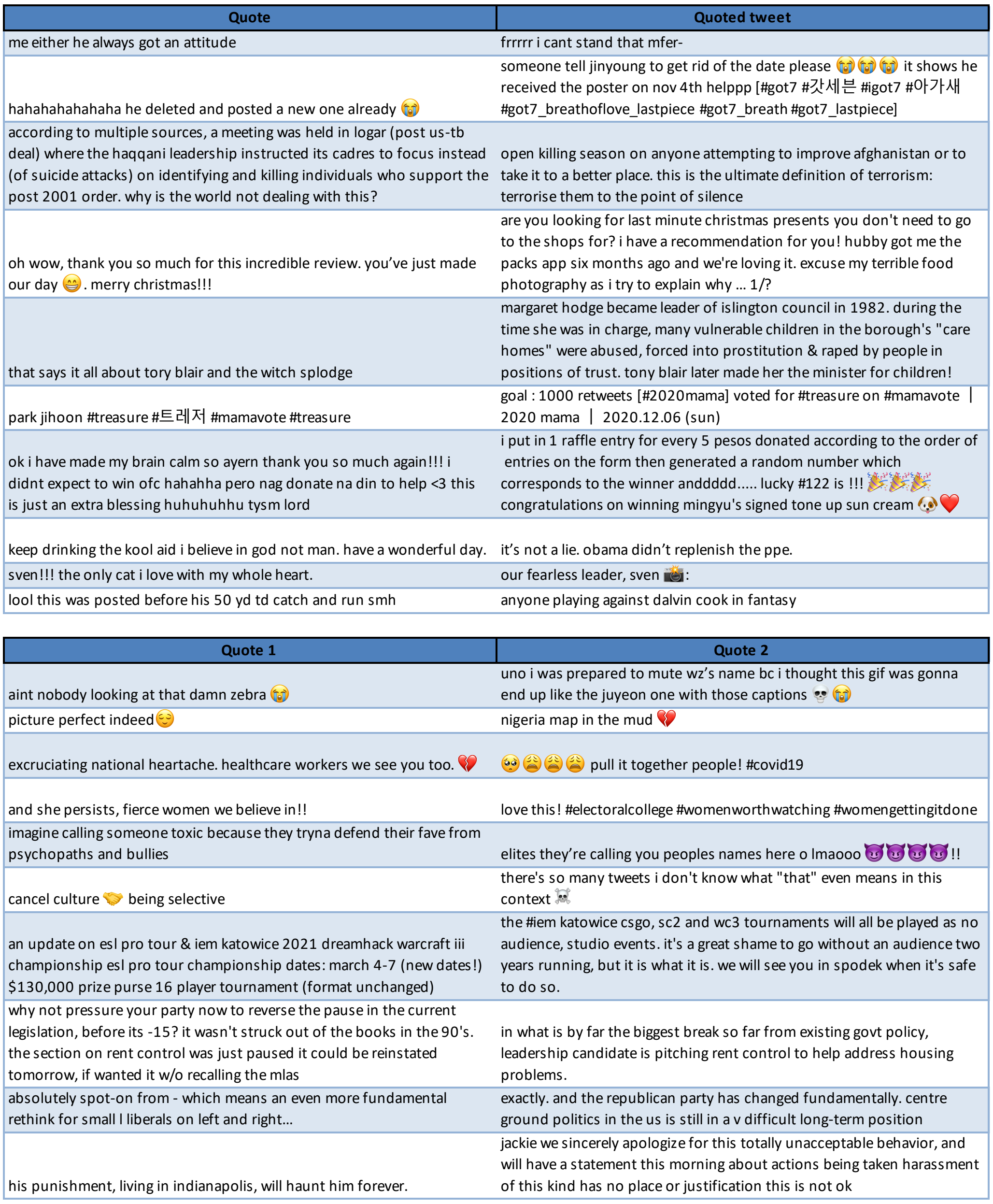
In this chapter I collect some detailed tables related to Section 4.2, Section 4.3 and Section 5.1.

Table A.1: *Datasets statistics, the first group represents controversial topics, while the second one represents non-controversial ones*

Hashtag/Keywords	#Lang	#Tweets	Description and collection period
#LeadersDebate	EN	250 000	Candidates debate, Nov 11-21,2019
pelosi	EN	252 000	Trump Impeachment, Dec 06,2019
@mauriciomacri	ES	108 375	Macri's mentions, Jan 1-11,2018
@mauriciomacri	ES	120 000	Macri's mentions, Mar 11-18,2018
@mauriciomacri	ES	147 709	Macri's mentions, Mar 20-27,2018
@mauriciomacri	ES	309 603	Macri's mentions, Apr 05-11,2018
@mauriciomacri	ES	254 835	Macri's mentions, May 05-11,2018
Kavanaugh	EN	260 000	Kavanaugh's nomination, Oct 03,2018
Kavanaugh	EN	259 999	Kavanaugh's nomination, Oct 05,2018
Kavanaugh	EN	260 000	Kavanaugh's nomination, Oct 08,2018
Bolsonaro	PT	170 764	Brazilian elections, Oct 27,2018
Bolsonaro	PT	260 000	Brazilian elections, Oct 28,2018
Bolsonaro	PT	260 000	Brazilian elections, 30-10-2018
Lula	PT	250 000	Mentions to Lula the day of Moro chats news, Jun 11-10,2019
Dilma	PT	209 758	Roussef impeachment, 06-11-2015
EXODEUX	EN	179 908	EXO's new album, Nov 07,2019
Thanksgiving	EN	250 000	Thanksgiving day, Nov 28,2019
#Al-HilalEntertainment	AR	221 925	Al-Hilal champion, Dec 01,2019
#MiracleOfChristmasEve	KO	251 974	Segun Woo singer birthday, 23-12-2019
Feliz Natal	PT	305 879	Happy Christmas wishes, Dec 24,2019
#kingjacksonday	EN	186 263	popstar's birthday, Mar 24-27,2019
#Wrestlemania	EN	260 000	Wrestlemania event, Apr 08,2019
Notredam	FR	200 000	Notredam fire, Apr 16,2019
Nintendo	EN	203 992	Nintendo's release, May 19-28,2019
Halsey	EN	250 000	Halsey's concert, Jun 07-08,2019
Bigil	EN	250 000	Vijay's birthday, Jun 21-22,2019
#VanduMuruganAJITH	EN	250 000	Ajith's fans, Jun 23,2019
Messi	ES	200 000	Messi's birthday, Jun 24,2019
#Area51	EN	178 220	Jokes about Area51, Jul 13,2019
#OTDirecto20E	ES	148 061	Event of a Music TV program in Spain, Jan 20,2020

Table A.2: *Most recurrent proper nouns in the vocabulary of 20 elected members of the Italian parliament, ranked by their frequency.*

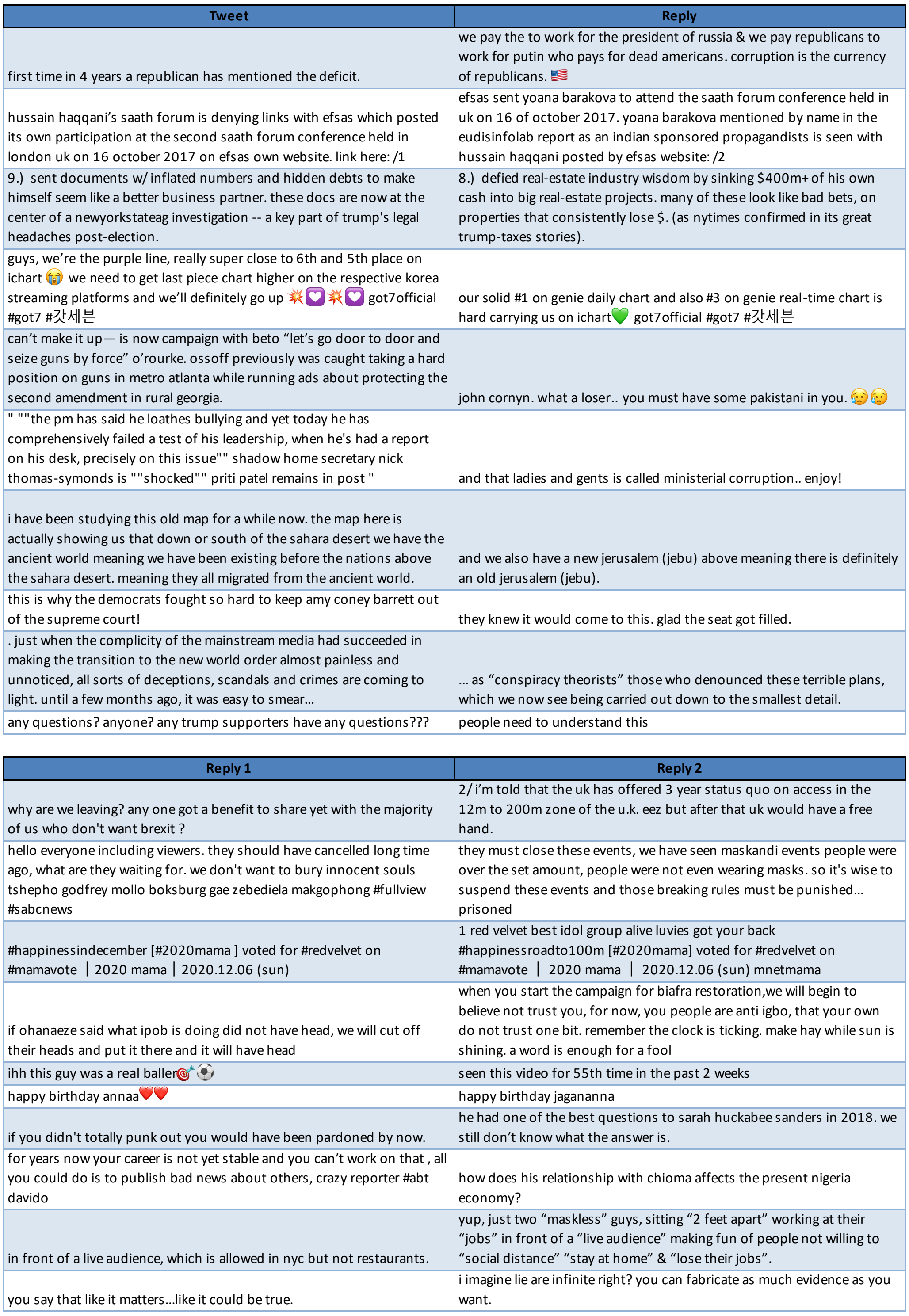
Democratic Party NNP	Frequencies	Right Parties NNP	Frequencies	Cinque Stelle NNP	Frequencies
Italia	0.085716	Italia	0.108296	Roma	0.069347
Bologna	0.049067	Europa	0.043148	Italia	0.042250
Roma	0.025675	Roma	0.033982	Cittá	0.026740
San	0.018526	Lazio	0.032541	Luigi	0.025314
Europa	0.014398	Liguria	0.021148	San	0.020323
Milano	0.011444	Forza	0.017809	Berlusconi	0.019966
Calabria	0.011142	San	0.014928	Piazza	0.018094
Berlusconi	0.009397	Friuli	0.014535	Torino	0.015955
Venezia	0.008994	Laura	0.014535	Augusta	0.015242
Forza	0.008390	Franco	0.013226	Sala	0.013459



Quote	Quoted tweet
me either he always got an attitude	frrrrr i cant stand that mfer-
hahahahahahaha he deleted and posted a new one already 😭	someone tell jinyoung to get rid of the date please 😭😭😭 it shows he received the poster on nov 4th helppp [#got7 #갓세븐 #igot7 #아가새 #got7_breathoflove_lastpiece #got7_breath #got7_lastpiece]
according to multiple sources, a meeting was held in logar (post us-tb deal) where the haqqani leadership instructed its cadres to focus instead (of suicide attacks) on identifying and killing individuals who support the post 2001 order. why is the world not dealing with this?	open killing season on anyone attempting to improve afghanistan or to take it to a better place. this is the ultimate definition of terrorism: terrorise them to the point of silence
oh wow, thank you so much for this incredible review. you've just made our day 😁. merry christmas!!!	are you looking for last minute christmas presents you don't need to go to the shops for? i have a recommendation for you! hubby got me the packs app six months ago and we're loving it. excuse my terrible food photography as i try to explain why ... 1/?
that says it all about tory blair and the witch splodge	margaret hodge became leader of islington council in 1982. during the time she was in charge, many vulnerable children in the borough's "care homes" were abused, forced into prostitution & raped by people in positions of trust. tony blair later made her the minister for children!
park jihoon #treasure #트레저 #mamavote #treasure	goal : 1000 retweets [#2020mama] voted for #treasure on #mamavote │ 2020 mama │ 2020.12.06 (sun)
ok i have made my brain calm so ayern thank you so much again!!! i didnt expect to win ofc hahahha pero nag donate na din to help <3 this is just an extra blessing huhuhuhhu tysm lord	i put in 1 raffle entry for every 5 pesos donated according to the order of entries on the form then generated a random number which corresponds to the winner anddddd..... lucky #122 is !!! 🎉🎉🎉 congratulations on winning mingyu's signed tone up sun cream 🐶❤️
keep drinking the kool aid i believe in god not man. have a wonderful day.	it's not a lie. obama didn't replenish the ppe.
sven!!! the only cat i love with my whole heart.	our fearless leader, sven 📸:
lool this was posted before his 50 yd td catch and run smh	anyone playing against dalvin cook in fantasy

Quote 1	Quote 2
aint nobody looking at that damn zebra 😭	uno i was prepared to mute wz's name bc i thought this gif was gonna end up like the juyeon one with those captions 💀😭
picture perfect indeed😌	nigeria map in the mud 💔
excruciating national heartache. healthcare workers we see you too. 💔	🥺😩😩😩 pull it together people! #covid19
and she persists, fierce women we believe in!!	love this! #electoralcollege #womenworthwatching #womengettingitdone
imagine calling someone toxic because they tryna defend their fave from psychopaths and bullies	elites they're calling you peoples names here o lmaooo 😈😈😈😈!!
cancel culture 🤝 being selective	there's so many tweets i don't know what "that" even means in this context ☠️
an update on esl pro tour & iem katowice 2021 dreamhack warcraft iii championship esl pro tour championship dates: march 4-7 (new dates!) $130,000 prize purse 16 player tournament (format unchanged)	the #iem katowice csgo, sc2 and wc3 tournaments will all be played as no audience, studio events. it's a great shame to go without an audience two years running, but it is what it is. we will see you in spodek when it's safe to do so.
why not pressure your party now to reverse the pause in the current legislation, before its -15? it wasn't struck out of the books in the 90's. the section on rent control was just paused it could be reinstated tomorrow, if wanted it w/o recalling the mlas	in what is by far the biggest break so far from existing govt policy, leadership candidate is pitching rent control to help address housing problems.
absolutely spot-on from - which means an even more fundamental rethink for small l liberals on left and right...	exactly. and the republican party has changed fundamentally. centre ground politics in the us is still in a v difficult long-term position
his punishment, living in indianapolis, will haunt him forever.	jackie we sincerely apologize for this totally unacceptable behavior, and will have a statement this morning about actions being taken harassment of this kind has no place or justification this is not ok

Figure A.1: *Examples of pairs of texts from Qt (top) and CoQt (bottom) datasets.*



Tweet	Reply
first time in 4 years a republican has mentioned the deficit.	we pay the to work for the president of russia & we pay republicans to work for putin who pays for dead americans. corruption is the currency of republicans. 🇺🇸
hussain haqqani's saath forum is denying links with efsas which posted its own participation at the second saath forum conference held in london uk on 16 october 2017 on efsas own website. link here: /1	efsas sent yoana barakova to attend the saath forum conference held in uk on 16 of october 2017. yoana barakova mentioned by name in the eudisinfolab report as an indian sponsored propagandists is seen with hussain haqqani posted by efsas website: /2
9.) sent documents w/ inflated numbers and hidden debts to make himself seem like a better business partner. these docs are now at the center of a newyorkstateag investigation -- a key part of trump's legal headaches post-election.	8.) defied real-estate industry wisdom by sinking $400m+ of his own cash into big real-estate projects. many of these look like bad bets, on properties that consistently lose $. (as nytimes confirmed in its great trump-taxes stories).
guys, we're the purple line, really super close to 6th and 5th place on ichart 😭 we need to get last piece chart higher on the respective korea streaming platforms and we'll definitely go up 💥💟💥💟 got7official #got7 #갓세븐	our solid #1 on genie daily chart and also #3 on genie real-time chart is hard carrying us on ichart💚 got7official #got7 #갓세븐
can't make it up— is now campaign with beto "let's go door to door and seize guns by force" o'rourke. ossoff previously was caught taking a hard position on guns in metro atlanta while running ads about protecting the second amendment in rural georgia.	john cornyn. what a loser.. you must have some pakistani in you. 😥😥
" ""the pm has said he loathes bullying and yet today he has comprehensively failed a test of his leadership, when he's had a report on his desk, precisely on this issue"" shadow home secretary nick thomas-symonds is ""shocked"" priti patel remains in post "	and that ladies and gents is called ministerial corruption.. enjoy!
i have been studying this old map for a while now. the map here is actually showing us that down or south of the sahara desert we have the ancient world meaning we have been existing before the nations above the sahara desert. meaning they all migrated from the ancient world.	and we also have a new jerusalem (jebu) above meaning there is definitely an old jerusalem (jebu).
this is why the democrats fought so hard to keep amy coney barrett out of the supreme court!	they knew it would come to this. glad the seat got filled.
. just when the complicity of the mainstream media had succeeded in making the transition to the new world order almost painless and unnoticed, all sorts of deceptions, scandals and crimes are coming to light. until a few months ago, it was easy to smear...	... as "conspiracy theorists" those who denounced these terrible plans, which we now see being carried out down to the smallest detail.
any questions? anyone? any trump supporters have any questions???	people need to understand this

Reply 1	Reply 2
why are we leaving? any one got a benefit to share yet with the majority of us who don't want brexit ?	2/ i'm told that the uk has offered 3 year status quo on access in the 12m to 200m zone of the u.k. eez but after that uk would have a free hand.
hello everyone including viewers. they should have cancelled long time ago, what are they waiting for. we don't want to bury innocent souls tshepho godfrey mollo boksburg gae zebediela makgophong #fullview #sabcnews	they must close these events, we have seen maskandi events people were over the set amount, people were not even wearing masks. so it's wise to suspend these events and those breaking rules must be punished... prisoned
#happinessindecember [#2020mama] voted for #redvelvet on #mamavote ｜2020 mama｜2020.12.06 (sun)	1 red velvet best idol group alive luvies got your back #happinessroadto100m [#2020mama] voted for #redvelvet on #mamavote ｜ 2020 mama ｜ 2020.12.06 (sun) mnetmama
if ohanaeze said what ipob is doing did not have head, we will cut off their heads and put it there and it will have head	when you start the campaign for biafra restoration,we will begin to believe not trust you, for now, you people are anti igbo, that your own do not trust one bit. remember the clock is ticking. make hay while sun is shining. a word is enough for a fool
ihh this guy was a real baller🎯⚽	seen this video for 55th time in the past 2 weeks
happy birthday annaa❤️❤️	happy birthday jagananna
if you didn't totally punk out you would have been pardoned by now.	he had one of the best questions to sarah huckabee sanders in 2018. we still don't know what the answer is.
for years now your career is not yet stable and you can't work on that , all you could do is to publish bad news about others, crazy reporter #abt davido	how does his relationship with chioma affects the present nigeria economy?
in front of a live audience, which is allowed in nyc but not restaurants.	yup, just two "maskless" guys, sitting "2 feet apart" working at their "jobs" in front of a "live audience" making fun of people not willing to "social distance" "stay at home" & "lose their jobs".
you say that like it matters...like it could be true.	i imagine lie are infinite right? you can fabricate as much evidence as you want.

Figure A.2: *Examples of pairs of texts from Rp (top) and CoRp (bottom) datasets.*

Table A.3: *Most recurrent nouns in the vocabulary of 20 elected members of the Italian parliament, ranked by their frequency. Nouns were translated from Italian to English by the authors.*

	Right Parties Nouns	**Frequencies**	**Cinque Stelle Nouns**	**Frequencies**	**Democratic Party Nouns**	**Frequencies**
0	government	0.020525	citizen	0.012416	job	0.014083
1	job	0.010293	job	0.010520	year	0.013420
2	year	0.010284	year	0.009318	government	0.012428
3	country	0.010215	law	0.009112	law	0.010318
4	right party	0.008931	government	0.008677	country	0.008362
5	brother	0.008686	star	0.008464	thing	0.007921
6	italian	0.008632	movement	0.007976	campaign	0.006723
7	president	0.008092	live	0.007611	day	0.006648
8	vote	0.007544	away	0.006767	person	0.006546
9	feature	0.007517	chamber	0.006494	citizen	0.005896
10	region	0.006502	country	0.006303	president	0.005836
12	tax	0.005896	program	0.005984	favour	0.005707
13	program	0.005862	president	0.005657	vote	0.005454
14	thing	0.005737	number	0.005653	woman	0.005443
15	citizen	0.005704	million	0.005204	club	0.005034
16	politics	0.005693	thing	0.005199	commitment	0.004850
17	security	0.005420	video	0.004862	hour	0.004712
18	day	0.005316	euro	0.004806	politics	0.004536
19	person	0.005312	city	0.004771	family	0.004435
20	state	0.005169	proposal	0.004529	program	0.004333

Table A.4: *Most relevant words (translated from Italian) per party as indicated by the coefficients of the linear regression model. The upper half of the table reports nouns that suggest the belonging to the party, the lower half the opposite*

	FI	Lega	M5S	PD
1	centre-right	lega	star	minister
2	president	people	movement	commitment
3	south	gazebo	citizen	suburbs
4	govern	north	change	thing
5	retired	right	spokesman	comparison
7834	thing	courtroom	family	citizen
7835	change	law	centre-right	people
7836	citizen	star	left	star
7837	star	president	minister	movement
7838	lega	govern	lega	centre-left

Bibliography

[1] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

[2] Leman Akoglu. Quantifying political polarity based on bipartite opinion networks. In *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[3] Mahmoud Al-Ayyoub, Abdullateef Rabab'ah, Yaser Jararweh, Mohammed N Al-Kabi, and Brij B Gupta. Studying the controversy in online crowds' interactions. *Applied Soft Computing*, 66:557–563, 2018.

[4] Righi Alessandra, Mauro M Gentile, and Domenico M Bianco. Who tweets in italian? demographic characteristics of twitter users. In *Convegno della Società Italiana di Statistica*, pages 329–344. Springer, 2017.

[5] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36, 2017.

[6] Gordon Willard Allport, Kenneth Clark, and Thomas Pettigrew. *The nature of prejudice*. Addison-wesley Reading, MA, 1954.

[7] Hind Almahmoud and Shurug AlKhalifa. Tsim: a system for discovering similar users on twitter. *Journal of Big Data*, 5, 10 2018.

[8] Reinald Kim Amplayo, Seonjae Lim, and Seung-won Hwang. Text length adaptation in sentiment classification. In Wee Sun Lee and Taiji Suzuki, editors, *Proceedings of The Eleventh Asian Conference on Machine Learning*, volume 101 of *Proceedings of Machine Learning Research*, pages 646–661, Nagoya, Japan, 17–19 Nov 2019. PMLR.

[9] Despoina Antonakaki, Paraskevi Fragopoulou, and Sotiris Ioannidis. A survey of twitter research: Data model, graph structure, sentiment analysis and attacks, 02 2021.

[10] David Anuta, Josh Churchin, and Jiebo Luo. Election bias: Comparing polls and twitter in the 2016 U.S. election. *CoRR*, 2017.

[11] Francesco Aquino, Gabriele Donzelli, Emanuela De Franco, Gaetano Privitera, Pier Luigi Lopalco, and Annalaura Carducci. The web and public confidence in mmr vaccination in italy. *Vaccine*, 35(35):4494–4498, 2017.

[12] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*, 2017.

[13] Sitaram Asur and Bernardo Huberman. Predicting the future with social media. *Proceedings - 2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010*, 1, 03 2010.

[14] Giuseppe Attardi. Wikiextractor. https://github.com/attardi/wikiextractor, 2015.

[15] Albert-László Barabási. *Network science*. Cambridge university press, 2016.

[16] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[17] Francesco Barbieri, Valerio Basile, Danilo Croce, Malvina Nissim, Nicole Novielli, and Viviana Patti. Overview of the evalita 2016 sentiment polarity classification task. *CEUR Workshop Proceedings*, pages 146–155, 01 2016.

[18] Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online, November 2020. Association for Computational Linguistics.

[19] Francesco Barbieri, German Kruszewski, Francesco Ronzano, and Horacio Saggion. How cosmopolitan are emojis? exploring emojis usage and meaning over different languages with distributional semantics. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, page 531–535, New York, NY, USA, 2016. Association for Computing Machinery.

[20] P. Basile, F. Cutugno, M. Nissim, V. Patti, and R. Sprugnoli. Evalita 2016: Overview of the 5th evaluation campaign of natural language processing and speech tools for italian. In *CEUR Workshop Proceedings*, volume 1749, 2016.

[21] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[22] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

[23] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, March 2003.

[24] Alessandro Bessi, Guido Caldarelli, Michela Del Vicario, Antonio Scala, and Walter Quattrociocchi. Social determinants of content selection in the age of (mis) information. In *International Conference on Social Informatics*, pages 259–268. Springer, 2014.

[25] Michael Beye, Arjan Jeckmans, Zekeriya Erkin, Pieter H. Hartel, Reginald Lagendijk, and Qiang Tang. *Literature Overview - Privacy in Online Social Networks*. Number TR-CTIT-10-36 in CTIT Technical Report Series. Centre for Telematics and Information Technology (CTIT), Netherlands, October 2010.

[26] Ali Bhatti, Muhammad Umer, Syed Adil, Mansoor Ebrahim, Daniyal Nawaz, and Faizan Ahmed. Explicit content detection system: An approach towards a safe and ethical environment. *Applied Computational Intelligence and Soft Computing*, 2018, 07 2018.

[27] David R Bild, Yue Liu, Robert P Dick, Z Morley Mao, and Dan S Wallach. Aggregate characterization of user behavior in twitter and analysis of the retweet graph. *ACM Transactions on Internet Technology (TOIT)*, 15(1), 2015.

[28] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

[29] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154 – 165, 2009.

[30] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[31] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[32] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[33] Johan Bollen, Huina Mao, and Alberto Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *CoRR*, 2011.

[34] Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Viégas, and Martin Wattenberg. An interpretability illusion for bert, 2021.

[35] Giovanni Bonaccorsi, Francesco Pierri, Matteo Cinelli, Francesco Porcelli, Alessandro Galeazzi, Andrea Flori, Ana Lucia Schmidth, Carlo Michele Valensise, Antonio Scala, Walter Quattrociocchi, and Fabio Pammolli. Economic and social consequences of human mobility restrictions under covid-19. *Proceedings of the National Academy of Sciences*, 2020.

[36] Stephen Borgatti and José Luis Molina. Ethical and strategic issues in organizational social network analysis. *The Journal of Applied Behavioral Science*, 39:337–349, 09 2003.

[37] Stephen P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55 – 71, 2005.

[38] Luc Bovens and Stephan Hartmann. *Bayesian Epistemology*. Number 9780199270408 in OUP Catalogue. Oxford University Press, 2004.

[39] Alexandre Bovet and Hernán A Makse. Influence of fake news in Twitter during the 2016 US presidential election. *Nature Communications*, 10(1):7, 2019.

[40] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[41] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[42] Danah Boyd, Scott Golder, and Gilad Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *2010 43rd Hawaii International Conference on System Sciences*, pages 1–10, 2010.

[43] danah m. boyd and Nicole B. Ellison. Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 10 2007.

[44] Marco Brambilla, Stefano Ceri, Florian Daniel, and Emanuele Della Valle. On the quest for changing knowledge. In *Proc. of the Workshop on Data-Driven Innovation on the Web DDI@WebSci, colocated with Web Science 2016, Hannover, Germany, May 22-25, 2016*, pages 3:1–3:5, 2016.

[45] Marco Brambilla, Stefano Ceri, Florian Daniel, Marco Di Giovanni, Andrea Mauri, and Giorgia Ramponi. Iterative knowledge extraction from social networks. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 1359–1364, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[46] Marco Brambilla, Stefano Ceri, Florian Daniel, Marco Di Giovanni, and Giorgia Ramponi. Content-based Community Detection and Characterization Dataset, 2019.

[47] Marco Brambilla, Stefano Ceri, Emanuele Della Valle, Riccardo Volonterio, and Felix Xavier Acero Salazar. Extracting emerging knowledge from social media. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 795–804, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[48] Giovanni Brena, Marco Brambilla, Stefano Ceri, Marco Di Giovanni, Francesco Pierri, and Giorgia Ramponi. News sharing user behaviour on twitter: A comprehensive data collection of news articles and social interactions. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 592–597, 2019.

[49] Markus Breunig, Hans-Peter Kriegel, Raymond Ng, and Joerg Sander. Lof: Identifying density-based local outliers. In *ACM Sigmod Record*, volume 29, pages 93–104, 06 2000.

[50] David A. Broniatowski, Amelia M. Jamison, SiHua Qi, Lulwah AlKulaib, Tao Chen, Adrian Benton, Sandra C. Quinn, and Mark Dredze. Weaponized health communication: Twitter bots and russian trolls amplify the vaccine debate. *American Journal of Public Health*, 108(10):1378–1384, 2018. PMID: 30138075.

[51] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[52] Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[53] Guido Caldarelli, Rocco De Nicola, Fabio Del Vigna, Marinella Petrocchi, and Fabio Saracco. The role of bot squads in the political propaganda on twitter. *arXiv preprint arXiv:1905.12687*, 2019.

[54] Mario Callegaro and Yongwei Yang. *The Role of Surveys in the Era of "Big Data"*, pages 175–192. Springer International Publishing, Cham, 2018.

[55] Ernesto Calvo. Anatomía política de twitter en argentina. *Tuiteando# Nisman. Buenos Aires: Capital Intelectual*, 2015.

[56] Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. Semantic re-tuning with contrastive tension. In *International Conference on Learning Representations*, 2021.

[57] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[58] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018.

[59] Julio Cesar Amador Diaz Lopez, Sofia Collignon-Delmar, Kenneth Benoit, and Akitaka Matsuo. Predicting the brexit vote by tracking and classifying public opinion using twitter data. *Statistics, Politics and Policy*, 8, 01 2017.

[60] Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. A multi-task approach for disentangling syntax and semantics in sentence representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2453–2464, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[61] Justin Cheng, Moira Burke, and Bethany de Gant. Country differences in social comparison on social media. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW3), January 2021.

[62] Qimin Cheng, Qian Zhang, Peng Fu, Conghuan Tu, and Sen Li. A survey and analysis on automatic image annotation. *Pattern Recognition*, 79:242 – 259, 2018.

[63] Cynthia Chew and Gunther Eysenbach. Pandemics in the age of twitter: content analysis of tweets during the 2009 h1n1 outbreak. *PloS one*, 5(11):e14118, 2010.

[64] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger,

Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

[65] Anderson Chris. *The long tail: Why the future of business is selling less of more*. New York: Hyperion, 2006.

[66] Alessandra Cignarella, Mirko Lai, Cristina Bosco, Viviana Patti, and Paolo Rosso. Sardistance @ evalita2020: Overview of the task on stance detection in italian tweets, 12 2020.

[67] Matteo Cinelli, Gianmarco De Francisci Morales, Alessandro Galeazzi, Walter Quattrociocchi, and Michele Starnini. Echo chambers on social media: A comparative analysis. *arXiv preprint arXiv:2004.09603*, 2020.

[68] Matteo Cinelli, Walter Quattrociocchi, Alessandro Galeazzi, Carlo Michele Valensise, Emanuele Brugnoli, Ana Lucia Schmidt, Paola Zola, Fabiana Zollo, and Antonio Scala. The COVID-19 Social Media Infodemic. *arXiv preprint arXiv:2003.05004*, pages 1–18, 2020.

[69] David Cohn and Thomas Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2001.

[70] Elanor Colleoni, Alessandro Rozza, and Adam Arvidsson. Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. *Journal of Communication*, 64(2):317–332, 2014.

[71] Alexis Conneau and Douwe Kiela. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[72] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[73] M. D. Conover, B. Goncalves, J. Ratkiewicz, A. Flammini, and F. Menczer. Predicting the political alignment of twitter users. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 192–199, Oct 2011.

[74] Michael D Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on twitter. In *Fifth international AAAI conference on weblogs and social media*, 2011.

[75] Alessandro Cossard, Gianmarco De Francisci Morales, Kyriaki Kalimeri, Yelena Mejova, Daniela Paolotti, and Michele Starnini. Falling into the echo chamber: The italian vaccination debate on twitter. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1):130–140, May 2020.

[76] Mathias Creutz. Open subtitles paraphrase corpus for six languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[77] CrowdTangle Team. CrowdTangle. Menlo Park, CA: Facebook., 2020. Accessed December 2020.

[78] Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. A comprehensive survey of multilingual neural machine translation, 2020.

[79] Jan-Willem Dam and Michel Velden. Online profiling and clustering of facebook users. *Decision Support Systems*, 70:60 – 72, 02 2015.

[80] Pranav Dandekar, Ashish Goel, and David T Lee. Biased assimilation, homophily, and the dynamics of polarization. *Proceedings of the National Academy of Sciences*, 110(15):5791–5796, 2013.

[81] Eleonora D'Andrea, Pietro Ducange, Alessio Bechini, Alessandro Renda, and Francesco Marcelloni. Monitoring the public opinion about the vaccination topic from tweets analysis. *Expert Systems with Applications*, 116:209–226, 2019.

[82] Kareem Darwish, Walid Magdy, and Tahar Zanouda. Trump vs. hillary: What went viral during the 2016 us presidential election. In Giovanni Luca Ciampaglia, Afra Mashhadi, and Taha Yasseri, editors, *Social Informatics*, pages 143–161, Cham, 2017. Springer International Publishing.

[83] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

[84] Juan Manuel Ortiz de Zarate, Marco Di Giovanni, Esteban Zindel Feuerstein, and Marco Brambilla. Measuring controversy in social networks through nlp. In Christina Boucher and Sharma V. Thankachan, editors, *String Processing and Information Retrieval*, pages 194–209, Cham, 2020. Springer International Publishing.

[85] Juan Manuel Ortiz de Zarate and Esteban Feuerstein. Vocabulary-based method for quantifying controversy in social media. *arXiv preprint arXiv:2001.09899*, 2020.

[86] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[87] Michela Del Vicario, Sabrina Gaito, Walter Quattrociocchi, Matteo Zignani, and Fabiana Zollo. News consumption during the italian referendum: A cross-platform analysis on facebook and twitter. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 648–657. IEEE, 2017.

[88] Michela Del Vicario, Gianna Vivaldo, Alessandro Bessi, Fabiana Zollo, Antonio Scala, Guido Caldarelli, and Walter Quattrociocchi. Echo chambers: Emotional contagion and group polarization on facebook. *Scientific Reports*, 6, 06 2016.

[89] Michela Del Vicario, Fabiana Zollo, Guido Caldarelli, Antonio Scala, and Walter Quattrociocchi. Mapping social dynamics on facebook: The brexit debate. *Social Networks*, 50:6–16, 2017.

[90] Matthew DeVerna, Francesco Pierri, Bao Truong, John Bollenbacher, David Axelrod, Niklas Loynes, Cristopher Torres-Lugo, Kai-Cheng Yang, Fil Menczer, and John Bryden. Covaxxy: A global collection of english twitter posts about covid-19 vaccines. *Proceedings of the International AAAI Conference on Web and Social Media*, 2021.

[91] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[92] Kaustubh D. Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahendiran, Simon Mille, Ashish Srivastava, Samson Tan, Tongshuang Wu, Jascha Sohl-Dickstein, Jinho D. Choi, Eduard Hovy, Ondrej Dusek, Sebastian Ruder, Sajant Anand, Nagender Aneja, Rabin Banjade, Lisa Barthe, Hanna Behnke, Ian Berlot-Attwell, Connor Boyle, Caroline Brun, Marco Antonio Sobrevilla Cabezudo, Samuel Cahyawijaya, Emile Chapuis, Wanxiang Che, Mukund Choudhary, Christian Clauss, Pierre Colombo, Filip Cornell, Gautier Dagan, Mayukh Das, Tanay Dixit, Thomas Dopierre, Paul-Alexis Dray, Suchitra Dubey, Tatiana Ekeinhor, Marco Di Giovanni, Rishabh Gupta, Rishabh Gupta, Louanes Hamla, Sang Han, Fabrice Harel-Canada, Antoine Honore, Ishan Jindal, Przemyslaw K. Joniak, Denis Kleyko, Venelin Kovatchev, Kalpesh Krishna, Ashutosh Kumar, Stefan Langer, Seungjae Ryan Lee, Corey James Levinson, Hualou Liang, Kaizhao Liang, Zhexiong Liu, Andrey Lukyanenko, Vukosi Marivate, Gerard de Melo, Simon Meoni, Maxime Meyer, Afnan Mir, Nafise Sadat Moosavi, Niklas Muennighoff, Timothy Sum Hon Mun, Kenton Murray, Marcin Namysl, Maria Obedkova, Priti Oli, Nivranshu Pasricha, Jan Pfister, Richard Plant, Vinay Prabhu, Vasile Pais, Libo Qin, Shahab Raji, Pawan Kumar Rajpoot, Vikas Raunak, Roy Rinberg, Nicolas Roberts, Juan Diego Rodriguez, Claude Roux, Vasconcellos P. H. S., Ananya B. Sai, Robin M. Schmidt, Thomas Scialom, Tshephisho Sefara, Saqib N. Shamsi, Xudong Shen, Haoyue Shi, Yiwen Shi, Anna Shvets, Nick Siegel, Damien Sileo, Jamie Simon, Chandan Singh, Roman Sitelew, Priyank Soni, Taylor Sorensen, William Soto, Aman Srivastava, KV Aditya Srivatsa, Tony Sun, Mukund Varma T, A Tabassum, Fiona Anting Tan, Ryan Teehan, Mo Tiwari, Marie Tolkiehn, Athena Wang, Zijian Wang, Gloria

Wang, Zijie J. Wang, Fuxuan Wei, Bryan Wilie, Genta Indra Winata, Xinyi Wu, Witold Wydmański, Tianbao Xie, Usama Yaseen, M. Yee, Jing Zhang, and Yue Zhang. Nl-augmenter: A framework for task-sensitive natural language augmentation, 2021.

[93] M. Di Giovanni and M. Brambilla. Efsg: Evolutionary fooling sentences generator. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 171–178, 2021.

[94] Marco Di Giovanni and Marco Brambilla. Content-based stance classification of tweets about the 2020 Italian constitutional referendum. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 14–23, Online, June 2021. Association for Computational Linguistics.

[95] Marco Di Giovanni and Marco Brambilla. Exploiting Twitter as source of large corpora of weakly similar pairs for semantic sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9902–9910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[96] Marco Di Giovanni, Marco Brambilla, Stefano Ceri, Florian Daniel, and Giorgia Ramponi. Content-based classification of political inclinations of twitter users. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4321–4327. IEEE, 2018.

[97] Marco Di Giovanni, Lorenzo Corti, Silvio Pavanetto, Francesco Pierri, Andrea Tocchetti, and Marco Brambilla. A content-based approach for the analysis and classification of vaccine-related stances on twitter: the italian scenario. *Workshop Proceedings of the 15th International AAAI Conference on Web and Social Media*, 2021.

[98] Joseph DiGrazia, Karissa Mckelvey, Johan Bollen, and Fabio Rojas. More tweets, more votes: Social media as a quantitative indicator of political behavior. *SSRN Electronic Journal*, 02 2013.

[99] William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[100] Gabriele Donzelli, Giacomo Palomba, Ileana Federigi, Francesco Aquino, Lorenzo Cioni, Marco Verani, Annalaura Carducci, and Pierluigi Lopalco. Misinformation on vaccination: a quantitative analysis of youtube videos. *Human vaccines & immunotherapeutics*, 14(7):1654–1659, 2018.

[101] Shiri Dori-Hacohen and James Allan. Automated controversy detection on the web. In *European Conference on Information Retrieval*, pages 423–434. Springer, 2015.

[102] DPCM. Ulteriori disposizioni attuative del decreto-legge 23 febbraio 2020, n. 6, recante misure urgenti in materia di contenimento e gestione dell'emergenza epidemiologica da covid-19. *Gazzetta Ufficiale*, 62(09-03-2020), 2020.

[103] Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. Self-training improves pre-training for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5408–5418, Online, June 2021. Association for Computational Linguistics.

[104] Eve Dubé, Caroline Laberge, Maryse Guay, Paul Bramadat, Réal Roy, and Julie A Bettinger. Vaccine hesitancy: an overview. *Human vaccines & immunotherapeutics*, 9(8):1763–1773, 2013.

[105] Susan T. Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38(1):188–230, 2004.

[106] Elizabeth Dwoskin. Twitter bans russian government-owned news sites rt and sputnik from buying ads. *The Washington Post*, 26-10-2017.

[107] David Easley, Jon Kleinberg, et al. *Networks, crowds, and markets*, volume 8. Cambridge university press Cambridge, 2010.

[108] Christopher Ifeanyi Eke, Azah Anir Norman, Liyana Shuib, and Henry Friday Nweke. A survey of user profiling: State-of-the-art, challenges, and solutions. *IEEE Access*, 7:144907–144924, 2019.

[109] Mohamed EL-MOUSSAOUI, Tarik AGOUTI, Abdessadek TIKNIOUINE, and Mohamed EL ADNANI. A comprehensive literature review on community detection: Approaches and applications. *Procedia Computer Science*, 151:295–302, 2019. The 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops.

[110] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. Open information extraction: The second generation. In *IJCAI*, pages 3–10. IJCAI/AAAI, 2011.

[111] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June 2008.

[112] Tom Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, June 2006.

[113] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[114] Wei Feng and Jianyong Wang. Retweet or not?: personalized tweet re-ranking. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 577–586. ACM, 2013.

[115] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Communications of the ACM*, 59(7):96–104, 2016.

[116] Antonietta Filia, Antonino Bella, Martina Del Manso, Melissa Baggieri, Fabio Magurano, and Maria Cristina Rota. Ongoing outbreak with well over 4,000 measles cases in italy from january to end august 2017- what is making elimination so difficult? *Eurosurveillance*, 22(37):30614, 2017.

[117] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[118] Fabio Franch. (wisdom of the crowds)2: 2010 uk election prediction with social media. *Journal of Information Technology & Politics*, 10(1):57–71, 2013.

[119] Isaac Chun-Hai Fung, Zion Tsz Ho Tse, Chi-Ngai Cheung, Adriana S Miu, and King-Wa Fu. Ebola and the social media. *The Lancet*, 2014.

[120] Alessandro Galeazzi, Matteo Cinelli, Giovanni Bonaccorsi, Francesco Pierri, Ana Lucia Schmidt, Antonio Scala, Fabio Pammolli, and Walter Quattrociocchi. Human mobility in response to covid-19 in france, italy and uk, 2020.

[121] Riccardo Gallotti, Francesco Valle, Nicola Castaldo, Pierluigi Sacco, and Manlio De Domenico. Assessing the risks of 'infodemics' in response to COVID-19 epidemics. *Nature Human Behaviour*, 4:1285–1293, 2020.

[122] F. Galton. Vox populi. *Nature*, 75(1949):7, 1907.

[123] Floriana Gargiulo, Florian Cafiero, Paul Guille-Escuret, Valérie Seror, and Jeremy Ward. Asymmetric participation of defenders and critics of vaccines to debates on french-speaking twitter. *Scientific Reports*, 10, 04 2020.

[124] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Reducing controversy by connecting opposing views. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 81–90. ACM, 2017.

[125] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Quantifying controversy on social media. *ACM Transactions on Social Computing*, 1(1):3, 2018.

[126] Daniel Gayo-Avello, Panagiotis Takis Metaxas, and Eni Mustafaraj. Limits of electoral predictions using twitter. In *ICWSM*, 2011.

[127] Shalmoli Ghosh, Prajwal Singhania, Siddharth Singh, Koustav Rudra, and Saptarshi Ghosh. Stance detection in web and social media: A comparative study. In Fabio Crestani, Martin Braschler, Jacques Savoy, Andreas Rauber, Henning Müller, David E. Losada, Gundula Heinatz Bürki, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 75–87, Cham, 2019. Springer International Publishing.

[128] Stamatios Giannoulakis and Nicolas Tsapatsoulis. Evaluating the descriptive power of instagram hashtags. *Journal of Innovation in Digital Ecosystems*, 3(2):114 – 129, 2016.

[129] Fabio Giglietto, Laura Iannelli, Luca Rossi, Augusto Valeriani, Nicola Righetti, Francesca Carabini, Giada Marino, Stefano Usai, and Elisabetta Zurovac. Mapping italian news media political coverage in the lead-up to 2018 general election. *Available at SSRN: https://ssrn.com/abstract=3179930*, 2018.

[130] John M. Giorgi, Osvald Nitski, Gary D. Bader, and Bo Wang. Declutr: Deep contrastive learning for unsupervised textual representations, 2020.

[131] Simone Giorgioni, Marcello Politi, Samir Salman, R. Basili, and Danilo Croce. Unitor @ sardistance2020: Combining transformer-based architectures and transfer learning for robust stance detection. In *EVALITA*, 2020.

[132] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[133] Ashish Goel, Aneesh Sharma, D. Wang, and Zhijun Yin. Discovering similar users on twitter. In *Workshop on mining and learning with graphs*, 2013.

[134] Sharad Goel, Ashton Anderson, Jake Hofman, and Duncan J Watts. The structural virality of online diffusion. *Management Science*, 62(1):180–196, 2015.

[135] Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

[136] Miha Grčar, Darko Cherepnalkoski, Igor Mozetič, and Petra Kralj Novak. Stance and influence of twitter users regarding the brexit referendum. *Computational social networks*, 4(1):6, 2017.

[137] Nir Grinberg, Kenneth Joseph, Lisa Friedland, Briony Swire-Thompson, and David Lazer. Fake news on twitter during the 2016 u.s. presidential election. *Science*, 363(6425):374–378, 2019.

[138] Tom Gruber. Collective knowledge systems: Where the social web meets the semantic web. *Web semantics: science, services and agents on the World Wide Web*, 6(1):4–13, 2008.

[139] Stefano Guarino, Francesco Pierri, Marco Di Giovanni, and Alessandro Celestini. Information disorders during the covid-19 infodemic: The case of italian facebook. *Online Social Networks and Media*, 22:100124, 2021.

[140] Stefano Guarino, Noemi Trino, Alessandro Celestini, Alessandro Chessa, and Gianni Riotta. Characterizing networks of propaganda on twitter: a case study. *arXiv preprint arXiv:2005.10004*, 2020.

[141] Stefano Guarino, Noemi Trino, Alessandro Chessa, and Gianni Riotta. Beyond fact-checking: Network analysis tools for monitoring disinformation in social media. In Hocine Cherifi, Sabrina Gaito, José Fernendo Mendes, Esteban Moro, and Luis Mateus Rocha, editors, *Complex Networks and Their Applications VIII*, pages 436–447, Cham, 2020. Springer International Publishing.

[142] Pedro Calais Guerra, Wagner Meira Jr, Claire Cardie, and Robert Kleinberg. A measure of polarization on social media networks based on community boundaries. In *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[143] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Zadeh. Wtf: The who to follow service at twitter. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 505–514, New York, NY, USA, 2013. Association for Computing Machinery.

[144] Poonam Gupta and Vishal Gupta. A survey of text question answering techniques. *International Journal of Computer Applications*, 53:1–8, 09 2012.

[145] Vishal Gupta and Gurpreet Lehal. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, 2, 08 2010.

[146] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Short Papers*, NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, pages 107–112. Association for Computational Linguistics (ACL), 2018.

[147] Lars Kai Hansen, Adam Arvidsson, Finn Aarup Nielsen, Elanor Colleoni, and Michael Etter. Good friends, bad news - affect and virality in twitter. In James J. Park, Laurence T. Yang, and Changhoon Lee, editors, *Future Information Technology*, pages 34–43, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[148] Mohammad Hasan and Mohammed Zaki. A survey of link prediction in social networks. *Soc Netw Data Anal*, pages 243–275, 03 2011.

[149] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[150] Irina Heimbach, Benjamin Schiller, Thorsten Strufe, and Oliver Hinz. Content virality on online social networks: Empirical evidence from twitter, facebook, and google+ on german news websites. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, HT '15, page 39–47, New York, NY, USA, 2015. Association for Computing Machinery.

[151] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient natural language response suggestion for smart reply. *ArXiv e-prints*, 2017.

[152] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California, June 2016. Association for Computational Linguistics.

[153] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In Aasa Feragen, Marcello Pelillo, and Marco Loog, editors, *Similarity-Based Pattern Recognition*, pages 84–92, Cham, 2015. Springer International Publishing.

[154] Sounman Hong. Online news on twitter: Newspapers' social media adoption and their online readership. *Information Economics and Policy*, 24(1):69–74, 2012.

[155] Dirk Hovy. Demographic factors improve classification performance. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 752–762, Beijing, China, July 2015. Association for Computational Linguistics.

[156] Philip N Howard and Bence Kollanyi. Bots,# strongerin, and# brexit: computational propaganda during the uk-eu referendum. *arXiv preprint arXiv:1606.06356*, 2016.

[157] James Y. Huang, Kuan-Hao Huang, and Kai-Wei Chang. Disentangling semantics and syntax in sentence embeddings with pre-trained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1372–1379, Online, June 2021. Association for Computational Linguistics.

[158] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PloS one*, 9(6):e98679, 2014.

[159] Myungha Jang. Probabilistic models for identifying and explaining controversy. *Doctoral Dissertation*, 2019.

[160] Myungha Jang, John Foley, Shiri Dori-Hacohen, and James Allan. Probabilistic approaches to controversy detection. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 2069–2072, 2016.

[161] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[162] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: Understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, WebKDD/SNA-KDD '07, pages 56–65, New York, NY, USA, 2007. ACM.

[163] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, pages 137–142, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[164] Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. Automatic sarcasm detection: A survey. *ACM Comput. Surv.*, 50(5), September 2017.

[165] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[166] Charles Kadushin. Who benefits from network analysis: ethics of social network research. *Social Networks*, 27(2):139–153, 2005. Ethical Dilemmas in Social Network Research.

[167] Gloria J. Kang, Sinclair R. Ewing-Nelson, Lauren Mackey, James T. Schlitt, Achla Marathe, Kaja M. Abbas, and Samarth Swarup. Semantic network analysis of vaccine sentiment in online social media. *Vaccine*, 35(29):3621–3638, 2017.

[168] George Karypis and Vipin Kumar. Metis—a software package for partitioning unstructured graphs, partitioning meshes and computing fill-reducing ordering of sparse matrices, 01 1997.

[169] Aditya Khosla, Atish Das Sarma, and Raffay Hamid. What makes an image popular? In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, page 867–876, New York, NY, USA, 2014. Association for Computing Machinery.

[170] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[171] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[172] G. S. Krishnan and S. S. Kamath. Dynamic and temporal user profiling for personalized recommenders using heterogeneous data sources. In *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–7, 2017.

[173] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[174] Dilek Küçük and Fazli Can. Stance detection: A survey. *ACM Comput. Surv.*, 53(1), February 2020.

[175] Juhi Kulshrestha, Muhammad Bilal Zafar, Lisette Espin Noboa, Krishna P Gummadi, and Saptarshi Ghosh. Characterizing information diets of social media users. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[176] Srijan Kumar, William L Hamilton, Jure Leskovec, and Dan Jurafsky. Community interaction and conflict on the web. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 933–943. International World Wide Web Conferences Steering Committee, 2018.

[177] Andrey Kupavskii, Liudmila Ostroumova, Alexey Umnov, Svyatoslav Usachev, Pavel Serdyukov, Gleb Gusev, and Andrey Kustarev. Prediction of retweet cascade size over time. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2335–2338. ACM, 2012.

[178] Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. Measuring bias in contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 166–172, Florence, Italy, August 2019. Association for Computational Linguistics.

[179] Michael LaCour. A balanced news diet, not selective exposure: Evidence from a direct measure of media exposure. In *APSA 2012 Annual Meeting Paper*, 2015.

[180] Preethi Lahoti, Kiran Garimella, and Aristides Gionis. Joint non-negative matrix factorization for learning ideological leaning on twitter. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 351–359. ACM, 2018.

[181] Mirko Lai, Viviana Patti, Giancarlo Ruffo, and Paolo Rosso. Stance evolution and twitter interactions in an italian political debate. In Max Silberztein, Faten Atigui, Elena Kornyshova, Elisabeth Métais, and Farid Meziane, editors, *Natural Language Processing and Information Systems*, pages 15–27, Cham, 2018. Springer International Publishing.

[182] Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. A continuously growing dataset of sentential paraphrases. In *Proceedings of The 2017 Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 1235–1245. Association for Computational Linguistics, 2017.

[183] David Lazer, Matthew Baum, Yochai Benkler, Adam Berinsky, Kelly Greenhill, et al. The science of fake news. *Science*, 359(6380):1094–1096, 2018.

[184] David M. J. Lazer, Matthew A. Baum, Yochai Benkler, Adam J. Berinsky, Kelly M. Greenhill, Filippo Menczer, Miriam J. Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, Michael Schudson, Steven A. Sloman, Cass R. Sunstein, Emily A. Thorson, Duncan J. Watts, and Jonathan L. Zittrain. The science of fake news. *Science*, 359(6380):1094–1096, 2018.

[185] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China, 22–24 Jun 2014. PMLR.

[186] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Chris Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 2014.

[187] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 631–640, New York, NY, USA, 2010. Association for Computing Machinery.

[188] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online, November 2020. Association for Computational Linguistics.

[189] Jing Li, Lingling Zhang, Fan Meng, and Fenhua Li. Recommendation algorithm based on link prediction and domain knowledge in retail transactions. *Procedia Computer Science*, 31:875 – 881, 2014. 2nd International Conference on Information Technology and Quantitative Management, ITQM 2014.

[190] Lei Li, Wei Peng, Saurabh Kataria, Tong Sun, and Tao Li. Recommending users and communities in social media. *ACM Trans. Knowl. Discov. Data*, 10(2):17:1–17:27, October 2015.

[191] Hui Lin and Vincent Ng. Abstractive summarization: A survey of the state of the art. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:9815–9822, 07 2019.

[192] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[193] C. Llewellyn and L. Cram. Brexit? analyzing opinion on the uk-eu referendum within twitter. In *ICWSM*, 2016.

[194] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*, 2018.

[195] Alessandro Lovari, Valentina Martino, and Nicola Righetti. Blurred shots: Investigating the information crisis around vaccination in italy. *American Behavioral Scientist*, 65(2):351–370, 2021.

[196] Alexander Maedche. *Ontology learning for the semantic web*, volume 665. Springer Science & Business Media, 2012.

[197] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining, 2018.

[198] Matt Mahoney. Large text compression benchmark, 2009.

[199] Eugenio Martínez-Cámara, Maria Martín-Valdivia, L. López, and Arturo Montejo-Ráez. Sentiment analysis in twitter. *Natural Language Engineering*, 20:1–28, 01 2014.

[200] Antonis Matakos, Evimaria Terzi, and Panayiotis Tsaparas. Measuring and moderating opinion polarization in social networks. *Data Mining and Knowledge Discovery*, 31(5):1480–1505, 2017.

[201] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093 – 1113, 2014.

[202] Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *International semantic web conference*, pages 522–536. Springer, 2005.

[203] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[204] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[205] Alan Mislove, Sune Lehmann, Yong-Yeol Ahn, Jukka-Pekka Onnela, and (James) Rosenquist. Understanding the demographics of twitter users. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, volume 11, 01 2011.

[206] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, pages 29–42, 01 2007.

[207] Stefano Mizzaro, Marco Pavan, and Ivan Scagnetto. Content-based similarity of twitter users. In Allan Hanbury, Gabriella Kazai, Andreas Rauber, and Norbert Fuhr, editors, *Advances in Information Retrieval*, pages 507–512, Cham, 2015. Springer International Publishing.

[208] Stefano Mizzaro, Marco Pavan, Ivan Scagnetto, and Martino Valenti. Short text categorization exploiting contextual enrichment and external knowledge. In *Proceedings of the First International Workshop on Social Media Retrieval and Analysis*, SoMeRA '14, page 57–62, New York, NY, USA, 2014. Association for Computing Machinery.

[209] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. A dataset for detecting stance in tweets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3945–3952, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[210] Paola Monachesi and Thomas Markus. Using social media for ontology enrichment. In *Extended Semantic Web Conference*, pages 166–180. Springer, 2010.

[211] Marçal Mora-Cantallops, Salvador Sánchez-Alonso, and Anna Visvizi. The influence of external political events on social networks: the case of the brexit twitter network. *Journal of Ambient Intelligence and Humanized Computing*, 03 2019.

[212] AJ Morales, Javier Borondo, Juan Carlos Losada, and Rosa M Benito. Measuring political polarization: Twitter shows the two sides of venezuela. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(3):033114, 2015.

[213] Sean A Munson, Stephanie Y Lee, and Paul Resnick. Encouraging reading of diverse political viewpoints with a browser widget. In *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[214] Seth A. Myers, Aneesh Sharma, Pankaj Gupta, and Jimmy Lin. Information network or social network? the structure of the twitter follow graph. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, page 493–498, New York, NY, USA, 2014. Association for Computing Machinery.

[215] Ramesh Nallapati and William W. Cohen. Link-plsa-lda: A new unsupervised model for topics and influence of blogs. In *ICWSM*, 2008.

[216] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108, Los Angeles, California, June 2010. Association for Computational Linguistics.

[217] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[218] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online, October 2020. Association for Computational Linguistics.

[219] Rasmus Kleis Nielsen, Nic Newman, Richard Fletcher, and Antonis Kalogeropoulos. Reuters institute digital news report 2019. *Report of the Reuters Institute for the Study of Journalism*, 2019.

[220] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. *International AAAI Conference on Weblogs and Social Media*, 11, 01 2010.

[221] Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. In Piotr Bański, Adrien Barbaresi, Hanno Biber, Evelyn Breiteneder, Simon Clematide, Marc Kupietz, Harald Lüngen, and Caroline Iliadi, editors, *Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019*, pages 9 – 16, Mannheim, 2019. Leibniz-Institut für Deutsche Sprache.

[222] Evelien Otte and Ronald Rousseau. Social network analysis: a powerful strategy, also for the information sciences. *Journal of Information Science*, 28(6):441–453, 2002.

[223] Mert Ozer, Nyunsu Kim, and Hasan Davulcu. Community Detection in Political Twitter Networks using Nonnegative Matrix Factorization Methods. *arXiv preprint arXiv:1608.01771*, 2016.

[224] Sowmya P and Madhumita Chatterjee. Detection of fake and cloned profiles in online social networks, 03 2019.

[225] Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3):515–554, May 2012.

[226] Raghavendra Reddy Pappagari, Piotr Żelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. Hierarchical transformers for long document classification. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844, 2019.

[227] Namkee Park, Seungyoon Lee, and Jang Hyun Kim. Individuals' personal network characteristics and patterns of facebook use: A social network approach. *Computers in Human Behavior*, 28:1700–1707, 09 2012.

[228] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[229] Yulong Pei, Nilanjan Chakraborty, and Katia Sycara. Nonnegative matrix tri-factorization with graph regularization for community detection in social networks. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2083–2089. AAAI Press, 2015.

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

[231] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[232] Thomas F Pettigrew and Linda R Tropp. Does intergroup contact reduce prejudice? recent meta-analytic findings. In *Reducing prejudice and discrimination*, pages 103–124. Psychology Press, 2013.

[233] Francesco Pierri. The diffusion of mainstream and disinformation news on twitter: the case of italy and france. *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion)*, 2020.

[234] Francesco Pierri, Alessandro Artoni, and Stefano Ceri. Investigating italian disinformation spreading on twitter in the context of 2019 european elections. *PloS one*, 15(1):e0227821, 2020.

[235] Francesco Pierri and Stefano Ceri. False news on social media: a data-driven survey. *ACM Sigmod Record*, 48(2), 2019.

[236] Francesco Pierri, Brea Perry, Matthew R DeVerna, Kai-Cheng Yang, Alessandro Flammini, Filippo Menczer, and John Bryden. The impact of online misinformation on us covid-19 vaccinations. *arXiv preprint arXiv:2104.10635*, 2021.

[237] Francesco Pierri, Carlo Piccardi, and Stefano Ceri. A multi-layer approach to disinformation detection in us and italian news spreading on twitter. *EPJ Data Science*, 9(35), 2020.

[238] Francesco Pierri, Carlo Piccardi, and Stefano Ceri. Topology comparison of Twitter diffusion networks effectively reveals misleading news. *Scientific Reports*, 10:1372, 2020.

[239] Francesco Pierri, Andrea Tocchetti, Lorenzo Corti, Marco Di Giovanni, Silvio Pavanetto, Marco Brambilla, and Stefano Ceri. Vaccinitaly: monitoring italian conversations around vaccines on twitter and facebook, 2021.

[240] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[241] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[242] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

[243] Ashwin Rajadesingan and Huan Liu. Identifying users with opposing opinions in twitter debates. In *International conference on social computing, behavioral-cultural modeling, and prediction*, pages 153–160. Springer, 2014.

[244] Anand Rajaraman and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA, 2011.

[245] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[246] Giorgia Ramponi, Marco Brambilla, Stefano Ceri, Florian Daniel, and Marco Di Giovanni. Vocabulary-based community detection and characterization. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, pages 1043–1050, New York, NY, USA, 2019. Association for Computing Machinery.

[247] Giorgia Ramponi, Marco Brambilla, Stefano Ceri, Florian Daniel, and Marco Di Giovanni. Content-based characterization of online social communities. *Information Processing & Management*, 57(6):102133, 2020.

[248] Gerasimos Razis and Ioannis Anagnostopoulos. Discovering similar twitter accounts using semantics. *Eng. Appl. Artif. Intell.*, 51(C):37–49, May 2016.

[249] Thomas Rebele, Fabian M. Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *ISWC 2016*, pages 177–185, 2016.

[250] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[251] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[252] Nils Reimers and Iryna 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, 11 2019.

[253] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2020.

[254] Nicola Righetti. Health politicization and misinformation on twitter. a study of the italian twittersphere from before, during and after the law on mandatory vaccinations, Apr 2020.

[255] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 399–408, New York, NY, USA, 2015. ACM.

[256] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.

[257] Yiye Ruan, David Fuhry, and Srinivasan Parthasarathy. Efficient community detection in large networks using content and links. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1089–1098, New York, NY, USA, 2013. ACM.

[258] Sebastian Ruder. Neural transfer learning for natural language processing, 2019.

[259] Gert Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.

[260] Mrinmaya Sachan, Danish Contractor, Tanveer A. Faruquie, and L. Venkata Subramaniam. Using content and interactions for discovering communities in social networks. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 331–340, New York, NY, USA, 2012. ACM.

[261] Facundo Sapienza and Pablo Groisman. Distancia de fermat y geodesicas en percolacion euclidea:teoriaa y aplicaciones en machine learning. *Msc Thesis*, 2018.

[262] Helmut Schmid. Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50, 1995.

[263] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015.

[264] Francesco Scotti, Davide Magnanimi, Valeria Maria Urbano, and Francesco Pierri. Online feelings and sentiments across italy during pandemic: investigating the influence of socio-economic and epidemiological variables. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 453–459. IEEE, 2020.

[265] Cristina Segalin, Dong Seon Cheng, and Marco Cristani. Social profiling through image understanding: Personality inference using convolutional neural networks. *Computer Vision and Image Understanding*, 156, 10 2016.

[266] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. The spread of low-credibility content by social bots. *Nature Communications*, 9:4787, 2018.

[267] Elisa Shearer and Jeffrey Gottfried. News use across social media platforms 2017. *Pew Research Center*, 7, 2017.

[268] Amit Sheth, Christopher Thomas, and Pankaj Mehra. Continuous semantics to analyze real-time data. *IEEE Internet Computing*, 14(6):84, 2010.

[269] Lei Shi, Neeraj Agarwal, Ankur Agrawal, Rahul Garg, and Jacob Spoelstra. Predicting us primary elections with twitter, 2012.

[270] Niloufar Shoeibi, Nastaran Shoeibi, Pablo Chamoso, Zakie Alizadehsani, and Juan Corchado Rodríguez. Similarity approximation of twitter profiles, 06 2021.

[271] Kuldeep Singh, Harish Kumar Shakya, and Bhaskar Biswas. Clustering of people in social network based on textual similarity. *Perspectives in Science*, 8:570 – 573, 2016. Recent Trends in Engineering and Material Sciences.

[272] Shashank Sheshar Singh, Ajay Kumar, Kuldeep Singh, and Bhaskar Biswas. C2im: Community based context-aware influence maximization in social networks. *Physica A: Statistical Mechanics and its Applications*, 514:796 – 818, 2019.

[273] Shashank Sheshar Singh, Kuldeep Singh, Ajay Kumar, and Bhaskar Biswas. Coim: Community-based influence maximization in social networks. In Ashish Kumar Luhach, Dharm Singh, Pao-Ann Hsiung, Kamarul Bin Ghazali Hawari, Pawan Lingras, and Pradeep Kumar Singh, editors, *Advanced Informatics for Computing Research*, pages 440–453, Singapore, 2019. Springer Singapore.

[274] A. Singhal. Introducing the knowledge graph: things, not strings. Available online at http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html, 2012.

[275] Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. A dataset for multi-target stance detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 551–557, Valencia, Spain, April 2017. Association for Computational Linguistics.

[276] Alessandro Spelta, Andrea Flori, Francesco Pierri, Giovanni Bonaccorsi, and Fabio Pammolli. After the lockdown: simulating mobility, public health and economic recovery scenarios. *Scientific reports*, 10(1):1–13, 2020.

[277] Leo G Stewart, Ahmer Arif, and Kate Starbird. Examining trolls and polarization with a retweet network. In *Proc. ACM WSDM, Workshop on Misinformation and Misbehavior Mining on the Web*, 2018.

[278] Stefan Stieglitz and Linh Dang-Xuan. Political communication and influence through microblogging–an empirical analysis of sentiment in twitter messages and retweet behavior. In *2012 45th Hawaii International Conference on System Sciences*, pages 3500–3509, 2012.

[279] L. Venkata Subramaniam, Shourya Roy, Tanveer A. Faruquie, and Sumit Negi. A survey of types of text noise and techniques to handle noisy text. In *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*, AND ’09, page 115–122, New York, NY, USA, 2009. Association for Computing Machinery.

[280] E. Sun and V. Iyer. Under the hood: The entities graph. Available online at https://www.facebook.com/notes/facebook-engineering/under-the-hood-the-entities-graph/10151490531588920/, 2013.

[281] Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert, 2020.

[282] P. Sun, X. Yang, X. Zhao, and Z. Wang. An overview of named entity recognition. In *2018 International Conference on Asian Language Processing (IALP)*, pages 273–278, 2018.

[283] Ke Tao, Fabian Abel, Qi Gao, and Geert-Jan Houben. Tums: Twitter-based user modeling service. In Raúl García-Castro, Dieter Fensel, and Grigoris Antoniou, editors, *The Semantic Web: ESWC 2011 Workshops*, pages 269–283, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[284] M. Taulé, M. Martí, Francisco M. Rangel Pardo, P. Rosso, C. Bosco, and V. Patti. Overview of the task on stance and gender detection in tweets on catalan independence. In *IberEval@SEPLN*, 2017.

[285] M. Taulé, Francisco M. Rangel Pardo, M. Martí, and P. Rosso. Overview of the task on multimodal stance detection in tweets on catalan #1oct referendum. In *IberEval@SEPLN*, 2018.

[286] CrowdTangle Team. Crowdtangle. menlo park, ca: Facebook. available at: https://www.crowdtangle.com, 2020.

[287] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Commun. ACM*, 59(2):64–73, January 2016.

[288] Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[289] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528, 2011.

[290] Trang Tran and Mari Ostendorf. Characterizing the language of online communities and its relation to community reception. *arXiv preprint arXiv:1609.04779*, 2016.

[291] Damian Trilling. Two different debates? investigating the relationship between a political debate on tv and simultaneous comments on twitter. *Social science computer review*, 33(3):259–276, 2015.

[292] Andranik Tumasjan, Timm Sprenger, Philipp Sandner, and Isabell Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *Proceedings of the International AAAI Conference on Web and Social Media*, 4(1):178–185, May 2010.

[293] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models, 2019.

[294] Lucia Vadicamo, Fabio Carrara, Andrea Cimino, Stefano Cresci, Felice Dell'Orletta, Fabrizio Falchi, and Maurizio Tesconi. Cross-media learning for image sentiment analysis in the wild. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 308–317, Oct 2017.

[295] Jannis Vamvas and Rico Sennrich. X-stance: A multilingual multi-target dataset for stance detection, 2020.

[296] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245, 2014.

[297] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[298] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[299] Eleni Vathi, Georgios Siolas, and Andreas Stafylopatis. *Mining Interesting Topics in Twitter Communities*, volume 9329, pages 123–132. Springer, 01 2015.

[300] Tommaso Venturini, Mathieu Jacomy, and Pablo Jensen. What do we see when we look at networks. an introduction to visual network analysis and force-directed layouts. *An introduction to visual network analysis and force-directed layouts (April 26, 2019)*, 2019.

[301] Michela Del Vicario, Walter Quattrociocchi, Antonio Scala, and Fabiana Zollo. Polarization and fake news: Early warning of potential misinformation targets. *ACM Transactions on the Web (TWEB)*, 13(2):10, 2019.

[302] Claudia Wagner and Markus Strohmaier. The wisdom in tweetonomies: Acquiring latent conceptual structures from social awareness streams. In *Proceedings of the 3rd International Semantic Search Workshop*, page 6. ACM, 2010.

[303] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461, 2018.

[304] Kexin Wang, Nils Reimers, and Iryna Gurevych. Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning, 2021.

[305] Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. Structbert: Incorporating language structures into pre-training for deep language understanding, 2019.

[306] Yuxi Wang, Martin McKee, Aleksandra Torbica, and David Stuckler. Systematic literature review on the spread of health-related misinformation on social media. *Social Science & Medicine*, 240:112552, 2019.

[307] Gerhard Weikum and Martin Theobald. From information to knowledge: harvesting entities and relationships from web sources. In *PODS*, pages 65–76. ACM, 2010.

[308] Katrin Weller, Axel Bruns, Jean Burgess, Merja Mahrt, and Cornelius Puschmann. *Twitter and society*, volume 89. Peter Lang, 2014.

[309] John Wieting and Kevin Gimpel. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[310] John Wieting, Graham Neubig, and Taylor Berg-Kirkpatrick. A bilingual generative transformer for semantic sentence embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1581–1594, Online, November 2020. Association for Computational Linguistics.

[311] Georgia Wilberding, Kurt; Wells. Facebook's timeline: 15 years in. *The Wall Street Journal*, 4-02-2019.

[312] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[313] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[314] Han Xiao. bert-as-service. `https://github.com/hanxiao/bert-as-service`, 2018.

[315] Wei Xu, Chris Callison-Burch, and Bill Dolan. SemEval-2015 task 1: Paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11, Denver, Colorado, June 2015. Association for Computational Linguistics.

[316] B. Yang and S. Manandhar. Community discovery using social links and author-based sentiment topics. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 580–587, Aug 2014.

[317] Kai-Cheng Yang, Francesco Pierri, Pik-Mai Hui, David Axelrod, Christopher Torres-Lugo, John Bryden, and Filippo Menczer. The covid-19 infodemic: Twitter versus facebook. *Big Data & Society*, 2021. Forthcoming in the special issue "Studying the COVID-19 Infodemic at Scale".

[318] Kai-Cheng Yang, Christopher Torres-Lugo, and Filippo Menczer. Prevalence of Low-Credibility Information on Twitter During the COVID-19 Outbreak. *arXiv preprint arXiv:2004.14484*, 2020.

[319] Xiao Yang, Craig Macdonald, and Iadh Ounis. Using word embeddings in twitter election classification. *Information Retrieval Journal*, 21(2-3):183–207, 2018.

[320] Sarita Yardi and Danah Boyd. Dynamic debates: An analysis of group polarization over time on twitter. *Bulletin of science, technology & society*, 30(5):316–327, 2010.

[321] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

[322] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences, 2020.

[323] John Zarocostas. How to fight an infodemic. *The Lancet*, 395(10225):676, 2020.

[324] Ding Zhou, Eren Manavoglu, Jia Li, C. Lee Giles, and Hongyuan Zha. Probabilistic models for discovering e-communities. In *Proceedings of the 15th International Conference on World Wide Web*, WWW ’06, pages 173–182, New York, NY, USA, 2006. ACM.