Learning User Latent Attributes on Social Media

Binxuan Huang

CMU-ISR-20-105
May 2020

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Kathleen M. Carley (Chair)
Yulia Tsvetkov
Zico Kolter
Huan Liu (Arizona State University)

*Submitted in partial fulfillment for the degree of
Doctor of Philosophy in Societal Computing*

The research reported in this thesis has been supported in part by the ONR Awards N000141612324, N000141512797, N000141812108, and N000141812106, the Knight Foundation and the Center for Computational Analysis of Social and Organization Systems (CASOS). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

Keywords: Social Media, User Profiling, Machine Learning, Text Mining, Graph Mining

Abstract

In recent years, there is a growing interest in using social media to understand social phenomena. Researchers have demonstrated many important applications of using online social media to understand real world events, such as presidential election prediction, earthquake early detection, and disaster management. A social media site is mixed with different types of users, in terms of gender, location, ideology, and etc. Different types of users may have different motivations, different opinions towards certain topics, different resources at their disposal, different behaviors in events. If researchers want to understand what is happening on a social media site, it is important to know where a post comes from, who wrote this post, and which party the author belongs to. However, this information is not explicitly provided by users.

In this thesis, the goal is to predict users' latent attributes such as their locations, social identities, and political orientations. Thanks to the massive text data on social media, we can learn rich knowledge from text to predict users' attributes. In the meanwhile, text data from social media often comes with a significant amount of meta data. Furthermore, data from social networks also contains rich connection information, eg. mentioning, following. It is still a challenge task to combine text, meta data, user network together for user attributes prediction.

In this thesis, I approach user attributes prediction at three levels — single post, user timeline, graph-level classification. I start with a global location prediction system that uses one single tweet as input to learn one user's location. It utilizes location-related features in a tweet, such as text and user profile metadata. I extend the tweet-level prediction system to user-level, which combines multiple posts in one user's timeline. I demonstrate the effectiveness of this model on the task of user social identity classification. An improved user-level hierarchical location prediction model is also presented. In these described models, I mainly focus on learning user attributes from users themselves. In the next step, I consider social graph as additional information to improve performance. Users connected in a social network often show similarities in certain aspects, which is a well-known phenomenon called social homophily. Experiments demonstrate that combining a social graph dramatically improves the performance of our prediction system compared to the previous user-level method. As a case study of the attributes prediction system, I apply the method on a real world emergency event — the novel coronavirus outbreak starting from 2019. I demonstrate that we gain better understanding of the public conversation during this global emergency event.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

In recent years, there is a growing interest in using social media to understand social phenomena. Researchers have demonstrated many important applications of user activity understanding in online social media, such as presidential election prediction [110], earthquake early detection [100], and disaster management [25].

Commonly, a social media site is mixed with users with various attributes, in terms of gender, location, political affiliation, social roles, and etc. Different types of users may have different motivations, different opinions towards certain topics, different resources at their disposal, different behaviors in events. If researchers want to understand what is happening on a social media site, it is important to know where a post comes from, who wrote this post, and which party the author belongs to.

In this thesis, the goal is to predict users' latent attributes such as their locations, social identities, and political orientations. Many times, these attributes are not explicitly provided by users. For example, there are no ways for users to specify their ideologies on Twitter. Even though there is a location field in a Twitter user's profile, it is often empty or filled with noisy information unrelated with any geographical location [50]. However, many times online users would unavoidably provide indicative clues which are helpful for identifying their attributes in their posts. For example, from a post "food in cmu is delicious :)", we can infer that this user is probably a student living in Pittsburgh.

Thanks to the massive text data on social media, we can learn rich knowledge from text to predict users' attributes. In the meanwhile, text data from social media often comes with a significant amount of meta data. Take Twitter for example, a single tweet object contains one short text with multiple meta fields like posting time, tweet language, user's personal description, and etc. How to efficiently handle text data combined with meta information still needs to be considered. Furthermore, data from social networks also contains rich connection information, eg. mentioning, following. It is still a challenge task to combine text, meta data, user network together for user attributes prediction.

In this thesis, I approach user attributes prediction at three levels — single post, user timeline, graph-level classification. First, I present a global location prediction system that uses one single tweet as input to learn one user's location. It utilizes location-related features in a tweet, such as text and user profile metadata. Second, I extend the tweet-level prediction system to user-level, which combines multiple posts in one user's timeline. I will demonstrate the effectiveness of this model on the task of user social identity classification. An improved user-level hierarchical location prediction model will also be presented. In these described models, I mainly focus on learning user attributes from users themselves. In the next step, I consider social graph as additional information to improve performance. Users connected in a social network often show similarities in certain aspects, which is a well-known phenomenon called social homophily [73]. As a case study of the attributes prediction system, I apply the method on a real world emergency event — the novel coronavirus outbreak starting from 2019. I demonstrate that we gain better understanding of the public conversation during this global emergency event.

Unlike previous work, my approaches use neural network to learn rich text representations and combine various feature fields. With the graph-level classification framework, the performance is improved a lot. Though I mainly use tweet data to predict user attributes like location, social identity, the methodology can be easily extended to other platforms like Facebook and Weibo, as well as other characteristics, eg. gender and age.

1.1 Background

Web user attributes prediction or user profiling has long been studied in the literature. Typically, researchers first define several categories to describe users. Then they use machine learning methods to classify users into these predefined attribute categories. Early work first represents a research paper by normalized term frequency, then these term frequency vectors are feed into a k-Nearest Neighbour classifier to classify topics of these papers [74]. Thus a user's research interests can be inferred by using cumulative paper topics. Similarly, Godoy et al. also transform webpages into term vectors, but they use hierarchical clustering instead of classification to group users based on their web interests [39].

Later, various feature engineering based methods are proposed to improve the performance of user attributes prediction. Following these term frequency based methods, Shmueli-Scheuer et al. further apply feature selection method to choose the best feature combinations [104]. Rao et al. create several features like tweeting frequency and number of followers in addition to the term features for Twitter user demographic prediction [95]. Similarly, Pennacchiotti add topic features from an LDA model and sentiment features from a sentiment lexicon for Twitter user classification [82]. Great improvement has been achieved with these hand-crafted features.

In recent years, deep neural network based methods show great learning capacity for various machine learning tasks, eg. computer recognition [49], language modeling [31], text classification [30]. Different from previous research, I propose to use deep neural networks to get better representative features without heavy feature engineering. The methods proposed in this thesis can also be easily generalized to other knowledge domains. Although some previous work use network features to improve their performance [82, 2, 95], most of them only touch some simple network heuristics like number of followers, n-popular friends. These methods may not fully utilize the network structure information. In this work, I will incorporate graph neural networks that propagate features from network neighbourhoods to make a better decision.

1.2 Datasets

1.2.1 Datasets built in this thesis

Tweet-level Location Dataset

For tweet-level geolocation prediction, we collect geotagged tweets from the whole world between January 7, 2017 and Febuary 1, 2017. For each user appeared in this collection, we randomly select one tweet for each city that one user has visited before. There are 3,321,194 users and 4,645,692 tweets in total.

Table 1.1 Summaries about the tweet-level location prediction dataset. Numbers in brackets are standard deviation.

# of tweets	# of users	# of timezones	# of lang.	# of countries (or regions)	Tweets per country	# of cities	Tweets per city
4645692	3321194	417	103	243	19118.0 (99697.1)	3709	1252.5(4184.5)

Table 1.2 A brief summary of our two identity datasets.

	Public figure		Identity						
	Verified	Unverified	Media	Reporter	Celebrity	Government	Company	Sport	Regular
Train	152368	365749	1140	614	876	844	879	870	6623
Dev.	1452	3548	52	23	38	40	35	43	269
Test	2926	7074	97	39	75	81	66	74	568

Identity Datasets

We build two datasets from Twitter — public figure dataset, identity dataset. In our first public figure dataset, we use Twitter's verification as a proxy for public figures, and these verified accounts include users in music, government, sports, business, and etc[1]. We sampled 156746 verified accounts and 376371 unverified accounts through Twitter's sample stream data [2]. Then we collected their most recent 20 tweets from Twitter's API in November 2018. We randomly choose 5000 users as a development set and 10000 users as a test set. A brief summary of this dataset is shown in Table 1.2.

In addition, we introduce another human labeled identity dataset for fine-grained identity classification, which contains seven identity classes: "news media", "news reporter", "government official", "celebrity", "company", "sport", and "regular user". For each identity, we manually labelled thousands of Twitter users and collected their most recent 20 tweets for classification in November 2018. For the regular Twitter users, we randomly sampled them from the Twitter sample stream.

Political orientation dataset

We also compiled one large-scale political orientation dataset. We first identify several popular political figures' Twitter accounts, then collect followers of these accounts. Those followers who only follow democratic politics are labeled as liberal users, while those who only follow republican politics are labeled as conservative users. One thing to note is that we exclude these politics' accounts in the dataset to make the task non-trivial. This dataset is built in October 2019. It covers the most recent 200 tweets of each individual account as of October 2019.

Table 1.3 Statistics of the political orientation dataset

Dataset	# of users		
	Train	Dev.	Test
Politic	791K	99K	99K

[1] https://help.twitter.com/en/managing-your-account/about-twitter-verified-accounts

[2] https://developer.twitter.com/en/docs/tweets/sample-realtime/overview/GET_statuse_sample.html

1.2.2 Datasets released by other groups

User-level Location Datasets

For user-level location prediction, we adopt three commonly used benchmark datasets. They are Twitter-US, Twitter-World, and WNUT.

Twitter-US is a dataset compiled by Roller et al. [98]. It is based on tweets collected between September 4th and November 29th, 2011. It contains 429K training users, 10K development users, and 10K test users in North America. The ground truth location of each user is set to the first geotag of this user in the dataset.

Twitter-World is a Twitter dataset covering the whole world, with 1,367K training users, 10K development users, and 10K test users [45]. It is collected using Twitter's streaming API between September 21 2011 to February 29 2012. The ground truth location for each user is the center of the closest city to the first geotag of this user. Only English tweets are included in this dataset, which makes it more challenging for a global-level location prediction task.

We downloaded these two datasets from website [3]. Each user in these two datasets is represented by the concatenation of their tweets, followed by the geo-coordinates. We queried Twitter's API to add user metadata information to these two datasets in February 2019. We only get metadata for about 53% and 67% users in Twitter-US and Twitter-World respectively. Because of Twitter's privacy policy change, we could not get the time zone information anymore at the time of collection.

WNUT is released in the 2nd Workshop on Noisy User-generated Text [48]. The original user-level dataset consists of 1 million training users, 10K users in development set and test set each. The authors filtered geotagged tweets from 2013 to 2016 via archived data from Twitter Streaming API. Because of Twitter's data sharing policy, only tweet ids of training and development data are provided. We have to query Twitter's API to reconstruct the training and development dataset. We finished our data collection around August 2017. About 25% training and development users' data cannot be accessed at that time. The full anonymized test data is downloaded from the workshop website [4].

Table 1.4 shows a brief summary of these three user-level location prediction datasets we use here.

[3] https://github.com/afshinrahimi/geomdn

[4] https://noisy-text.github.io/2016/geo-shared-task.html

Table 1.4 A brief summary of our user-level location prediction datasets. For each dataset, we report the number of users, number of users with metadata, number of tweets, and average number of tweets per user. We collected metadata for 53% and 67% of users in Twitter-US and Twitter-World. Time zone information was not available when we collected meta data for these two datasets. About 25% of training and development users' data was inaccessible when we collected WNUT in 2017.

	Twitter-US			Twitter-World			WNUT		
	Train	Dev.	Test	Train	Dev.	Test	Train	Dev.	Test
# users	429K	10K	10K	1.37M	10K	10K	742K	7.46K	10K
# users with meta	228K	5.32K	5.34K	917K	6.50K	6.48K	742K	7.46K	10K
# tweets	36.4M	861K	831K	11.2M	488K	315K	8.97M	90.3K	99.7K
# tweets per user	84.60	86.14	83.12	8.16	48.83	31.59	12.09	12.10	9.97

1.3 Summary of Thesis

This thesis is composed by seven chapters:

- Chapter 1: Introduction
- Chapter 2: Tweet-level Location Prediction (published at SBP-BRiMS 2017 [54])
- Chapter 3: User-level Social Identity Classification (published at Disinformation, Misinformation, and Fake News in Social Media [58])
- Chapter 4: Hierarchical User Location Prediction (published at EMNLP 2019 [55])
- Chapter 5: Graph-level Attributes Prediction
- Chapter 6: An Empirical Study of the Novel Coronavirus Outbreak on Twitter

The first chapter introduces this thesis and provides some background information. I present some prior work on attributes prediction for web users. I also introduce datasets used in this thesis.

The second chapter presents one tweet-level location prediction system that can operate on the global-level with multi-lingual support. I present a new method to predict a Twitter user's location based on the information in a single tweet. This method integrates text and user profile meta-data into a single model. Our experiments demonstrate that our neural model substantially outperforms baseline methods, achieving 52.8% accuracy and 92.1% accuracy on city-level and country-level prediction respectively.

The third chapter provides a user-level prediction model for social identity classification on Twitter. It combines information available in users' timelines. We first collect a coarse-grained public figure dataset automatically, then manually label a more fine-grained identity dataset. We propose a hierarchical self-attention neural network for Twitter user role identity classification. Our experiments demonstrate that the proposed model significantly outperforms multiple baselines. We further propose a transfer learning scheme that improves our model's performance by a large margin and greatly reduces the need for a large amount of human labeled data.

In the fourth chapter, we propose a hierarchical location prediction neural network to improve Twitter user geolocation. It first predicts the home country for a user, then uses the country result to guide the city-level prediction. In addition, we employ a character-aware word embedding layer to overcome the noisy information in tweets. It not only improves the prediction accuracy but also greatly reduces the mean error distance.

In previous chapters, I have shown various methods for user attributes prediction. Most of them only use user's local features without any network information, which may contain useful information for prediction. In the fifth chapter, I combine social networks into my prediction models. Specifically, I use previous user-level architecture to extract user representations and build a graph attention network on top these user representation features. I further demonstrate that under semi-supervised setting we can greatly improve our system's performance by adding unlabeled users.

In chapter 6, I show a case study of the presented methods on a global emergency event — the outbreak of a novel coronavirus disease. With the help of the attributes prediction system, we are interested in four research questions: 1. Who are the sources of influential tweets in this global health emergency event? 2. Who are the sources of fake news and misinformation? 3. Which countries are the origins of fake news and misinformation? Does fake news distribute the same as real news geographically? 4. How does fake news and misinformation spread internationally?

Chapter 7 summarizes this thesis and provides some discussion.

Chapter 2

Tweet-level Location Prediction

2.1 Introduction

Recently, there is growing interest in using social media to understand social phenomena. For example, researchers have shown that analyzing social media reveals important geospatial patterns for keywords related to presidential elections[110]. People can use Twitter as a sensor to detect earthquakes in real-time[100]. Recent research also has demonstrated that Twitter data provides real-time assessments of flu activity[1].

Using Twitter's API[1], a keyword search can be done and we can easily get tweet streams from across the world containing keywords of interest. However, we cannot conduct a fine-grained analysis in a specific region using such a keyword-based search method. Alternatively, using the same API tweets with geo-information can be collected via a bounding box. Since less than 1% of tweets are tagged with geo-coordinates[43], using this location-based search means we will lose the majority of the data. If we can correctly locate those ungeotagged tweets returned from a keyword search stream, that would enable us to study users in a specific region with far more information.

With this motivation, we are aiming to study the problem of inferring a tweet's location. Specifically, we are trying to predict on a tweet by tweet basis, which country and which city it comes from. Most of the previous studies rely on rich user information(tweeting history and/or social ties), which is time-consuming to collect because of the Twitter API's speed limit. Thus those methods could not be directly applied to Twitter streams. In this paper, we study a global location prediction system working on each single tweet. One data sample is one tweet JSON object returned by Twitter's streaming API. Our system utilizes location-related features in a tweet, such

[1] https://dev.twitter.com/docs

as text and user profile meta-data. We summarize useful features that can provide information for location prediction in Table 2.1.

Table 2.1 Feature table for tweet location prediction.

Feature	Type
Tweet content	Free text
User personal description	Free text
User name	Free text
User profile location	Free text
Tweet language(TL)	Categorical
User language(UL)	Categorical
Timezone(TZ)	Categorical
Posting time(PT)	UTC timestamp

Recent research has shown that using bag-of-words and classical machine learning algorithms such as Naive Bayes can provide us a text-based location classifier with good accuracy[47]. Different from previous research, we intend to use the convolutional neural network(CNN) to boost prediction power. Inspired by the success of convolutional neural network in text classification[63], we are going to use CNN to extract location related features from texts and train a classifier that combines high-level text feature representations with these categorical features. To benchmark our method, we compared our approach with a stacking-based method. Experimental results demonstrate that our approach achieves 92.1% accuracy on country-level prediction and 52.8% accuracy on city-level prediction, which greatly outperforms our baseline methods on both tasks.

2.2 Related Work

There is increasing interest in inferring Twitter user's location, which is largely driven by the lack of sufficient geo-tagged data[43]. In many situations, it is important to know where a tweet came from in order to use the information in the tweet to effect a good social outcome. Key examples include: disaster relief[67], earthquake detection[36], and predicting flu trends[1].

A majority of previous works either focus on a local region e.g. United States[26], Sweden[10], or using rich user information like a certain number of tweets for each user[26], user's social relationship[62, 88]. Different from these works, this paper works on worldwide tweet location prediction. We only utilized features in one single tweet without any external information. Thus this method could be easily applied to real-time Twitter stream.

For fine-grained location prediction, there are several types of location representation methods existing in literature. One typical method is to divide earth into small grids and try to predict which cell one tweet comes from. Wing and Baldridge introduced a grid-based representation with fixed latitude and longitude[116]. Based on the similarity measured by Kullback-Leibler divergence, they assign each ungeotagged tweet to the cell with most similar tweets. Because cells in urban area tend to contain far more tweets than the ones in rural areas, the target classes are rather imbalanced[46]. To overcome this, Roller et al. further proposed an adaptive grid representation using K-D tree partition[98]. Another type of representation is topic region. Hong et al. proposed a topic model to discover the latent topic words for different regions[53]. Such parametric generative model requires a fixed number of regions. However, the granularity of topic regions is hard to control and will potentially vary over time[47].

The representation we choose is city-based representation considering most tweets come from urban area. One early work proposed by Cheng et al. using a probabilistic framework to estimate Twitter user's city-level location based on the content of tweets[26]. Their framework tries to identify local words with probability distribution smoothing. However, such method needs a certain number of tweets(100) for each user to get a good estimation. Han et al. proposed a stacking-based approach to predict user's city[46]. They combine tweet text and meta-data in user profile with stacking[117]. Specifically, they train a multinomial naive Bayes base classifier on tweet text, profile location, timezone. Then they train a meta-classifier over the base classifiers. More recently, Han et al. further did extensive experiments to show that using feature selection method, such as information gain ratio[90] could greatly improve the classification performance.

2.3 Tweet Location Prediction

In this section, we will introduce our location prediction approach. We first briefly describe the useful features in a tweet JSON object. After that, we will further explain how we utilize these features in our prediction model.

2.3.1 Feature Set

We have listed all useful information we want to utilize in Table 2.1. Tweet content, user personal description, user name and profile location are four text fields that we will use. Twitter users often reveal their home location in their profile location and personal description. However, location indicating words are often mixed with informal tweet text(e.g. chitown for Chicago). It is unrealistic to use a gazetteer to find these words. In this work, we choose to apply CNN on these four text fields to extract high-level representations.

In addition to these four text fields. there are another three categorical features: tweet language, user language,and timezone. Tweet language is automatically determined by Twitter's language detection tool. User language and timezone are selected by the user in his/her profile. These three categorical features are particularly useful for distinguishing users at the country-level.

The last feature is UTC posting time. Using posting timestamp as a discriminative feature is motivated by the fact that people in a region are more active on Twitter at certain times during the day. For example, while people in United Kingdom start to be active at 9:00 am in UTC time, most of the people in United States are still asleep. We transform the posting time in UTC timestamp into discrete time slots. Specifically, we divide 24 hours into 144 time slots each with a length of 10 minutes. Thus each tweet will have a discrete time slot number in the range of 144, which can be viewed as a categorical feature. In Figure 2.1, we plotted the probability distribution of an user posting tweets in each time slot in three different countries. As expected, there is a big variance between these three countries.

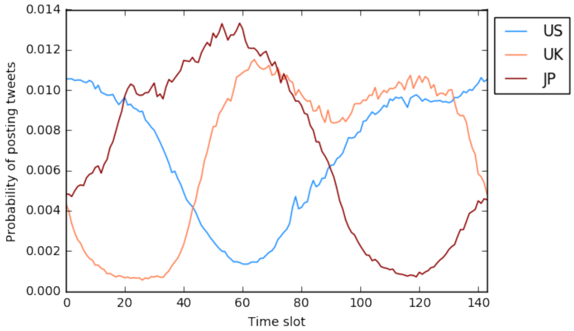


Fig. 2.1 The probability of an user posting a tweet in different time slot in three different countries: United States, United Kingdom, Japan.

2.3.2 Our Approach

Our approach is based on the convolutional neural network for sentence classification proposed by Kim[63]. Different from traditional bag-of-words method, such convolutional neural networks take the word order into consideration. Our model architecture is shown in Figure 2.2. We use this CNN architecture to extract high-level features from four text fields in a tweet. Let $x_i^t \in R^k$ be the k-dimensional word vector corresponding to the i-th word in the text t, where $t \in$ {tweet content, user description, profile location, user name}. As a result, one text of length n can be represented as a matrix

$$X_{1:n}^t = x_1^t \oplus x_2^t \oplus .... \oplus x_n^t \tag{2.1}$$

where $\oplus$ is concatenation operator. In the convolutional layer, we apply each filter $w \in R^{hk}$ to all the word vector matrices, where h is the window size and k is the length of a word vector. For example, applying filter w to a window of word vectors $x_{i:i+h-1}^t$, we generated $c_i^t = f(w \cdot x_{i:i+h-1}^t + b)$. Here $b \in R$ is a bias term and we choose $f(x)$ as a non-linear ReLU function $max(x,0)$. Sliding the filter window from the beginning of a word matrix till the end, we generated a feature vector $c^t = [c_1^t, c_2^t, ..., c_{n-h+1}^t]$ for each text t. If we have m filters in the convolutional layer, then we can produce m feature vectors for each text field and $4m$ vectors in total.

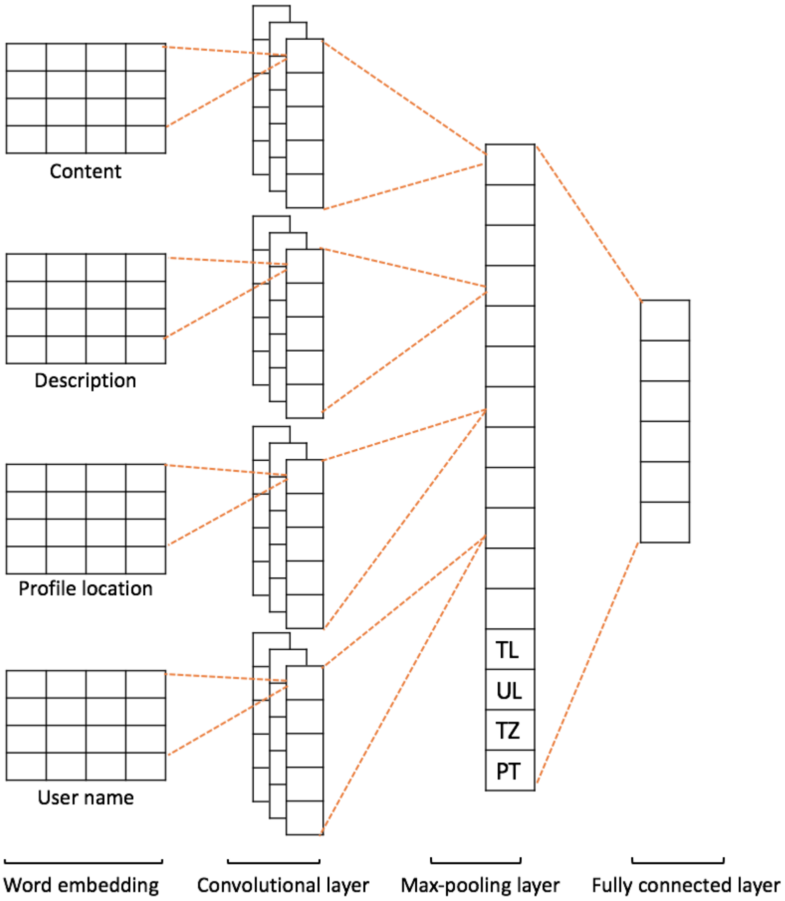


Fig. 2.2 Architecture of our tweet location prediction model.

In the max-pooling layer, we apply a pooling operation over each feature vector generated in the convolutional layer. Each pooling operation takes a feature vector as input and outputs the maximum value $\hat{c}^t = max(c^t)$. $\hat{c}^t$ can be viewed as the most representative feature generated by a filter on text t. Hence we finally got a long vector $\theta \in R^{4m}$ after the max-pooling layer. To avoid the co-adaptation of hidden units, we apply dropout on the max-pooling layer that randomly set elements in θ to zero in the training phase. After that, we append four categorical features tweet language(TL), user language(UL), timezone(TZ) and posting time(PT) with one-hot encoding at the end of θ and get $\hat{\theta}$. In the last fully connected layer, we use a softmax function over this long vector $\hat{\theta}$ to generate the probability distribution over locations. Specifically, the probability of one tweet coming from location l_i is

$$P(l_i|\hat{\theta}) = \frac{exp(\beta_i^T \hat{\theta})}{\sum_{j=1}^{L} exp(\beta_j^T \hat{\theta})} \tag{2.2}$$

where L is the number of locations and β_i are parameters in softmax layer. The output predicted location is just the location with highest probability.

The minimization objective in the training phase is the categorical cross-entropy loss. The parameters to be estimated include word vectors, weight vectors w for each filters, the weight vectors β in softmax layer, and all the bias terms. The optimization is performed using mini-batch stochastic gradient descent and back-propagation[99].

2.4 Data

We used geo-tagged tweets collected from Twitter streaming API[2] for training and evaluation. In this study, we set the geographic bounding box as [-180, -90, 180, 90] so that we could get these geo-tagged tweets from the whole world. Our collection started from January 7, 2017 to February 1, 2017. Because it is very common for one user to post tweets from the same city, we randomly chose one tweet for each city that one user has visited. This could ensure that there is no strong overlap among our data samples. We are only using tweets either with specific geo-coordinates or a geo-bounding box smaller than [0.1,0.1]. For the latter case, we are using the center of one tweet's bounding box as its coordinates. Besides this, we haven't done any data filtering to ensure our data sample is close to real world situation. There are 3,321,194 users and 4,645,692 tweets in total. We randomly selected 10% users' tweets

[2] https://dev.twitter.com/streaming/reference/post/statuses/filter

as testing data. For the remaining 90% users, we picked tweets from 50,000 of them as a development set and used the remaining tweets as training data.

There are two location prediction tasks we consider in this paper. The first task is country-level location prediction. We adopted the country code in the geo-tagged tweet as the label we want to predict. In our dataset, there are 243 countries and regions in total. The second task is city-level location classification. We adopt the same city-based representation as Han et al.[16]. The city-based representation consists of 3,709 cities throughout the world and is obtained by aggregating smaller cities with the largest nearby city. We assign the closest city for each tweet based on orthodromic distance. In Table 2.2 are some basic statistics about our dataset. It is worth mentioning that this dataset is rather imbalanced, where a majority of tweets are sent from a few countries/cities.

Table 2.2 Summaries about the dataset. Numbers in brackets are standard deviation.

# of tweets	# of users	# of timezones	# of lang.	# of countries (or regions)	Tweets per country	# of cities	Tweets per city
4645692	3321194	417	103	243	19118.0 (99697.1)	3709	1252.5(4184.5

2.5 Experiments

2.5.1 Evaluation Measures

Following previous work of tweet geolocation prediction[46], we used four evaluation measures listed below. One thing to note is that when we calculated the error distance we used distance between predicted city and the true coordinates in the tweet rather than the center of assigned closest city.

- Acc: The percentage of correct location predictions.
- Acc@Top5: The percentage of true location in our top 5 predictions.
- Acc@161: The percentage of predicted city which are within a 161km(100 mile) radius of the true coordinates in the original tweet to capture near-misses. This measure is only tested on city-level prediction.
- Median: The median distance from the predicted city to the true coordinates in the original tweet. This measure is only tested on city-level prediction.

2.5.2 Baseline Method

We compared our approach with one commonly used ensemble method in previous research works [46, 47]. We implemented an ensemble classifier based on stacking[117] with 5-fold cross validation. The training of stacking consists of two steps. First, five multinomial naive Bayes base classifiers are trained on different types of data(tweet content, user description, profile location, user name and the remaining categorical features). The outputs from the base classifiers are used to train a multinomial naive Bayes classifier in the second layer. We call such method STACKING in this paper. Same as [47], we also use information gain ratio to do feature selection on text tokens. We call STACKING with feature selection STACKING+.

2.5.3 Hyperparameters and Training

We used a tweet-specific tokenizer provided by NLTK to tokenize text fields. We built our dictionary based on the words that appeared in text, user description, and profile location. To reduce low-utility words and noise, we removed all words that had a word frequency less than 10. For our proposed approach, we used filter windows(h) of 3,4,5 with 128 feature vectors each, a dropout rate of 0.5 and batch size of 1024. We initialize word vectors using word2vec vectors trained on 100 billion words from Google News. The vectors have dimensionality of 300 and were trained using the continuous bag-of-words architecture[75]. For those words that are not included in word2vec, we initialized them randomly. We also performed early stopping based on the accuracy over the development set. Training was done through stochastic gradient descent using Adam update rule with learning rate 10^{-3}[64]. For our baseline models, we applied additive smoothing with $\alpha = 10^{-2}$, which is selected on the development set. For STACKING+ method, we first ranked these words by their information gain ratio value, then selected the top n% words as our vocabulary. The tuning of n is based on accuracy over the development set. We selected n as 40%, 55% for city-level prediction and country-level prediction respectively.

2.5.4 Results

The comparison results between our approach and the baseline methods are listed in Table 2.3. Our approach achieves 92.1% accuracy and 52.8% accuracy on country-level and city-level location prediction respectively. Our approach is consistently better than the previous model on the country-level location prediction task as shown in Table 2.3. It greatly outperforms our baseline methods over all the measures, especially on

Table 2.3 Country-level and city-level location prediction results.

	Country		City			
	Acc	Acc@Top5	Acc	Acc@161	Acc@Top5	Median
STACKING	0.868	0.947	0.389	0.573	0.595	77.5 km
STACKING+	0.871	0.950	0.439	0.616	0.629	47.2 km
Our approach	**0.921**	**0.972**	**0.528**	**0.692**	**0.711**	**28.0** km

the city-level prediction task. It could assign more than half of the test tweets to the correct city and gain more than 20% relative improvement over the accuracy of the STACKING+ method.

To show the importance of each type of feature, we do an ablation study where we remove each feature one by one. In Table 2.4, we show the performance of our model without each type of feature. Bar charts in Figure 2.3 show the performance loss in country-level and city-level prediction when removing each feature. The profile location is the most important feature for country-level location prediction. Surprisingly, tweet text is more important than profile location for predicting user's home city. Possible reason is that a lot of users provide meaningless or coarse-grained location in this field [56]. User's name is the least informative feature in both cases.

Table 2.4 Ablation study for location prediction.

	Country		City		
	Acc.	Acc@Top5	Acc.	Acc@Top 5	Acc@161
Full model	0.921	0.972	0.528	0.711	0.692
w/o text	0.856	0.939	0.332	0.555	0.549
w/o description	0.906	0.966	0.497	0.680	0.667
w/o location	0.830	0.942	0.400	0.557	0.540
w/o name	0.898	0.964	0.505	0.693	0.678
w/o categorical	0.885	0.955	0.498	0.679	0.662

Our approach performs better for countries with a large number of tweets. In Figure 2.4, we plotted the precision and recall value for each country as a scatter chart. The dot size is proportional to the number of tweets that come from that country. Turkey appears to be the country with highest precision and recall. These results suggest that our approach works better with more data samples.

The same graph is also plotted for city prediction in Figure 2.4. Because of the skewness of our data and the difficulty of city-level prediction, our classifier tends to generate labels towards big cities, which leads to high recall and low precision for cities like Los Angeles.

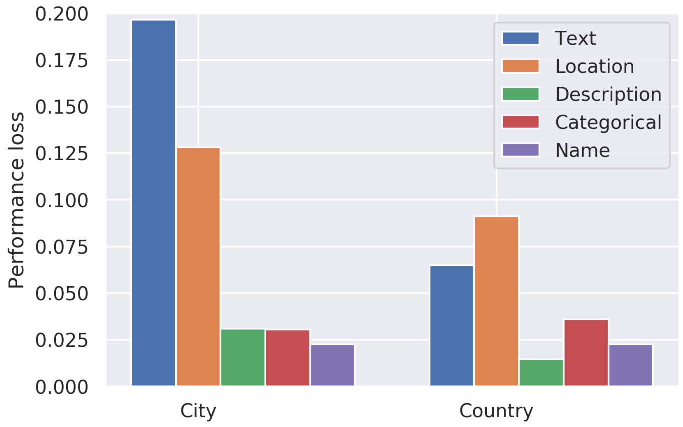


Fig. 2.3 Bar charts show the performance loss when removing each feature.

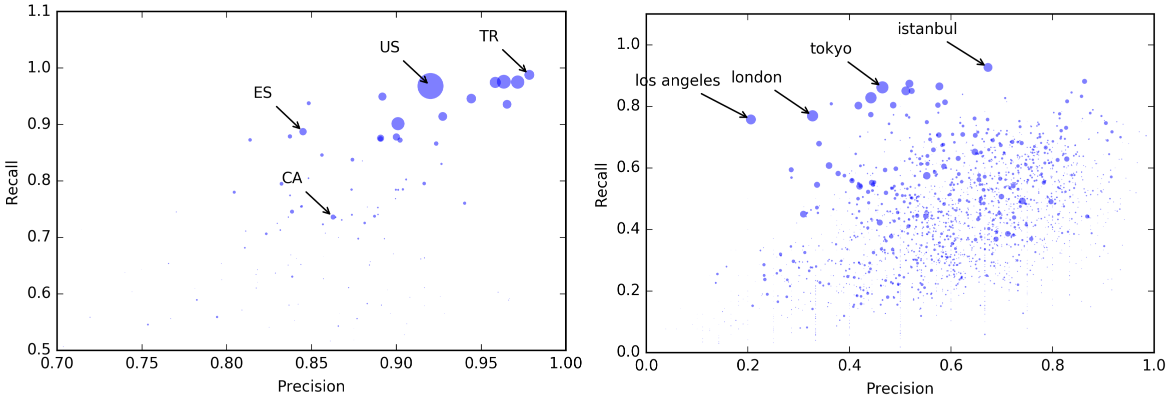


Fig. 2.4 Two scatter graphs that show the performances for each country and cities. The x-axis is precision, y-axis is recall. Each dot represents a country/city. The dot size is proportional to the number of tweets that comes from the correponding location. Some tiny invisible country outside of the scope are not shown in the figure.

For real world applications, people may ask how we could set a threshold to get prediction results with high confidence. To answer this question, we further examined the relation between prediction accuracy and the output probability. Here the output probability is just the probability of our predicted location calculated by equation 2.2. Figure 2.5 shows the distribution of tweets in terms of output probability for two tasks. As expected, the prediction accuracy increases as the output probability increases. We get 97.2% accuracy for country-level prediction with output probability larger than 0.9. Surprisingly, the accuracy of city-level is as high as 92.7% for the 29.6% of the tweets with output probability greater than 0.9. However, the city-level accuracy for the remaining tweets with output probability less than 0.9 is only 48.4%.

Unlike country-level prediction, the number of tweets decreases as output probability increases, unless the output probability is larger than 0.9.

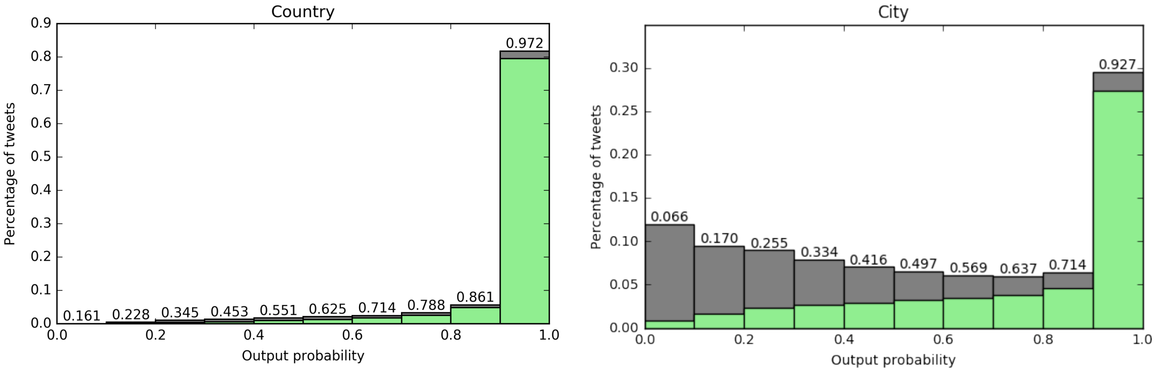


Fig. 2.5 Two bar charts that show the location probability distribution of tweets. Two bar charts that show the distribution of tweets in terms of the output probability. The x-axis is the output probability associated with each prediction, the y-axis is the percentage of tweets. The height of grey bar represents the percentage of test data that has certain output probability. The height of green bar represents the percentage of correctly predicted tweets in each probability range. We listed the accuracy for each probability range above the bar. Take the rightmost bar in country-level prediction for example, there are 81.8% tweets' country are predicted with output probability larger than 0.9. Among these 81.8% tweets, 97.2% are predicted correctly.

2.6 Case Study

In this section, we provide a case study to show how this location prediction system can help us to better understand a real world dataset. Specifically, we collected a Twitter dataset using keyword search [3]. The keyword we chose here is "ukraine". In total, there are 18297 tweets in this dataset, while only 292 of them are geotagged (1.6%). If we want to find out which countries are mainly involved in the discussion about Ukraine, only using these 292 tweets can barely provide us meaningful information.

In Figure 2.6, we show the number of geotagged tweets sent from each country in the left. As expected, Ukraine is the dominant country in the dataset because of the chosen keyword. Countries ranking the second and third are German and Russia. However, after we apply our country-level prediction system on this dataset, the country distribution changes dramatically. Again, Ukraine is the country with the most of tweets. But this time, United States and United Kingdom turn our to be the countries

[3] https://developer.twitter.com/en/docs/tweets/filter-realtime/guides/basic-stream-parameters

with the second/third largest amount of tweets. Based on our further analysis, we found certain accounts such as news agencies, eg. “JuliaDavisNews” annd “Newsweek” from United States mentioned Ukraine in their tweets while not using geotags.

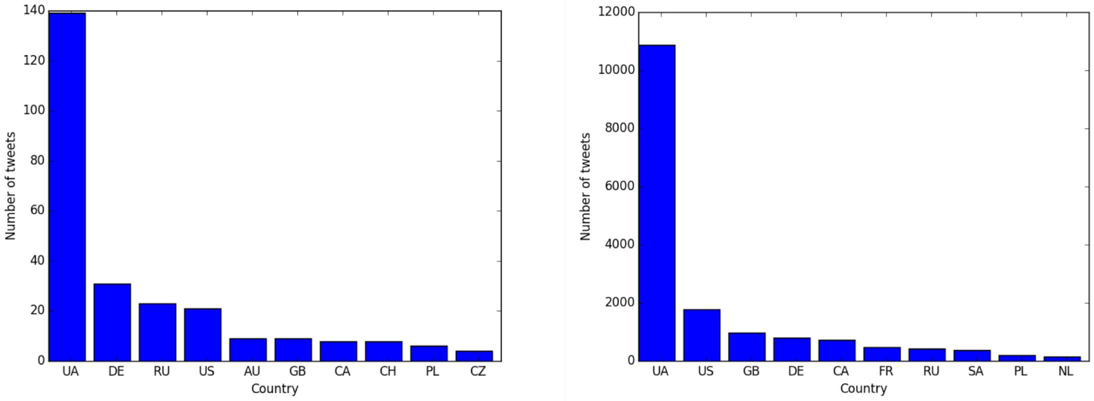


Fig. 2.6 Country distribution in the dataset. The left figure is the distribution before prediction, the right one is after prediction.

Similarly, in the country co-hashtag network, we can see there is very few countries and links in the co-hashtag network before applying location predictor. Ukraine is the central node in this graph. After location prediction, there are more conversations happening between United States and other countries and US turns out to be as important as UA in the conversation.

2.7 Discussion and Conclusion

These experiments demonstrate that our approach is consistently better than the prior method thus supporting more tweets to be accurately located by country, and in some cases by city, of origin. At the country level, the more tweets that come from the country, the better the prediction. Regardless of the number of tweets per country, we can predict the country location for most tweets wih extremely high confidence and accuracy. At the city level the results are more mixed. For a small fraction of tweets we can get greater than 90% accuracy, but for the rest of tweets the accuracy is less than 50%. For about half the tweets it is difficult to infer the city location. This result is partially due to the fact that we base the prediction on only a single tweet. Future work may consider using collection of tweets per user. This result is also partially due to the fact that the data is highly skewed toward a few cities. Future

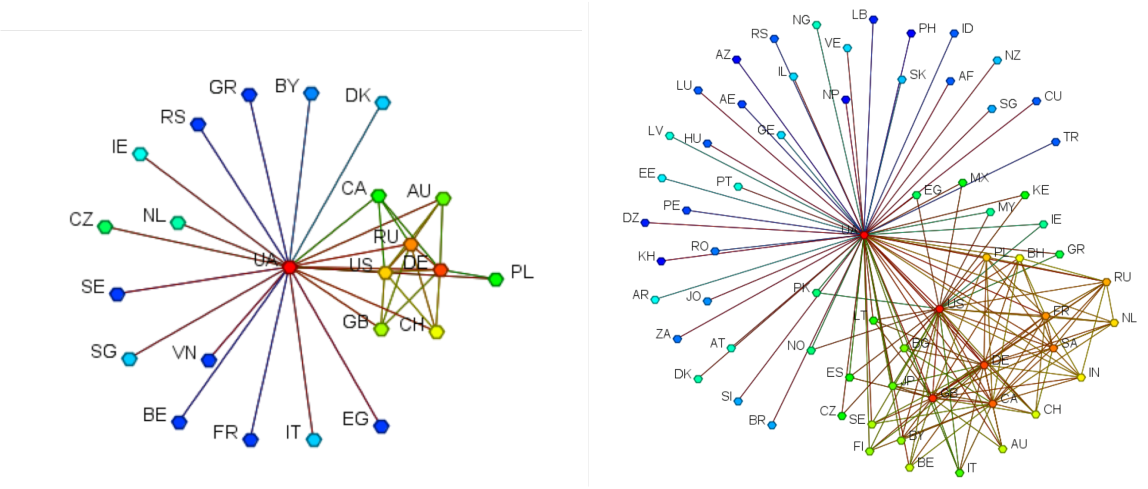


Fig. 2.7 A visualization of country co-hashtag network. If two countries share one common hashtag, then there is a link with weight one connecting these two countries. Nodes are colored by their total degrees. We hide the links which have weight lower than the mean value in the visualization. The left figure is the network before location predicting, the right one is after predicting.

work should develop a training set that is more evenly distributed across cities. Despite these limitations, this approach shows promise.

This paper presents a method for geo-locating a single tweet based on the information in a tweet JSON object. The proposed approach integrates tweet text and user profile meta-data into a single model. Compared to the previous stacking method with feature selection, our approach substantially outperforms the baseline method. We developed the approach for both city and country level and demonstrated the ability to classify almost 50% and 90% of all tweets at city-level and country-level respectively. The results demonstrate that using a convolutional neural network utilizes the textual location information better than previous approaches and boosts the location prediction performance substantially.

Chapter 3

User-level Social Identity Classification

3.1 Introduction

An identity is a characterization of the role an individual takes on. It is often described as the social context specific personality of an individual actor or a group of people [5]. Identities can be things like jobs (e.g. “lawyer”, “teacher”), gender (man, woman), or a distinguishing characteristic (e.g. “a shy boy”, “a kind man”). People with different identities tend to exhibit different behaviors in the social space [21]. In this paper, we use identity to refer to the roles individuals or groups play in society.

Specifically on social media platforms, there are many different kinds of actors using social media, e.g., people, organizations, and bots. Each type of actors has different motivations, different resources at their disposal, and may be under different internal policies or constraints on when they can use social media, how they can represent themselves, and what they can communicate. If we want to understand who is controlling the conversation and whom is being impacted, it is important to know what types of actors are doing what.

To date, for Twitter, most research has separated types of actors largely based on whether the accounts are verified by Twitter or not [51], or whether they are bots or not [27]. However, a variety of different types of actors may be verified - e.g., news agencies, entertainment or sports team, celebrities, and politicians. Similarly, bots can vary - e.g., news bots and non-news bots. If we could classify the identities of actors on Twitter, we could gain an improved understanding of who was doing the influencing and who was being influenced [24]. This would lead to improved accuracy in measuring the impact of marketing and influence campaigns.

In this paper, the primary goal is to classify Twitter users based on their identities on social media. First, we introduce two datasets for Twitter user identity classification. One is automatically collected from Twitter aiming at identifying public figures on social media. Another is a human labeled dataset for more fine-grained Twitter user identity classification, which includes identities like government officials, news reporters, etc. Second, we present a hierarchical self-attention neural network for Twitter user identity classification. In our experiments, we show our method achieves excellent results when compared to many strong classification baselines. Last but not least, we propose a transfer learning scheme for fine-grained user identity classification which boosts our model's performance a lot.

3.2 Related Work

Sociologists have long been interested in the usage of identities across various social contexts [107]. As summarized in [106], three relatively distinct usages of *identity* exist in the literature. Some use identity to refer to the culture of a people [20]. Some use it to refer to common identification with a social category [108]. While others use identity to refer to the role a person plays in highly differentiated contemporary societies. In this paper, we use the third meaning. Our goal for identity classification is to separate actors with different roles in online social media.

Identity is the way that individuals and collectives are distinguished in their relations with others [60]. Certain difficulties still exist for categorizing people into different groups based on their identities. Recasens et al. [96] argue that identity should be considered to be varying in granularity and a categorical understanding would limit us in a fixed scope. While much work could be done along this line, at this time we adopt a coarse-grained labeling procedure, that only looks at major identities in the social media space.

Twitter, a popular online news and social networking site, is also a site that affords interactive identity presentation to unknown audiences. As pointed out by Robinson [97], individuals form new cyber identities on the internet, which are not necessarily the way they would be perceived offline. A customized identity classifier is needed for online social media like Twitter.

A lot of research has tried to categorize Twitter users based on certain criteria, like gender [18], location [54], and political orientation [29]. Another similar research topic is bot detection [27], where the goal is to identify automated user accounts from normal Twitter accounts. Differing from them, our work tries to categorize Twitter users based on users' social identity or social roles. Similarly, Priante et al. [86] also study identity classification on Twitter. However, their approach is purely based on profile description, while we combine user self-description and tweets together. Additionally, we demonstrate that tweets are more helpful for identity classification than personal descriptions in our experiments.

In fact, learning Twitter users' identities can benefit other related tasks. Twitter is a social media where individual user accounts and organization accounts co-exist. Many user classification methods may not work on these organization accounts, e.g., gender classification. Another example is bot detection. In reality, accounts of news agencies and celebrities often look like bots [28], because these accounts often employ automated services or teams (so called cyborgs), and they also share features with certain classes of bots; e.g., they may be followed more than they follow. Being able to classify actors' roles on Twitter would improve our ability to automatically differentiate pure bots from celebrity accounts.

3.3 Method

In this section, we describe details of our hierarchical self-attention neural networks. The overall architecture is shown in Figure 3.1. Our model first maps each word into a low dimension word embedding space, then it uses a Bidirectional Long Short-Term Memory (Bi-LSTM) network [52] to extract context specific semantic representations for words. Using several layers of multi-head attention neural networks, it generates a final classification feature vector. In the following parts, we elaborate these components in details.

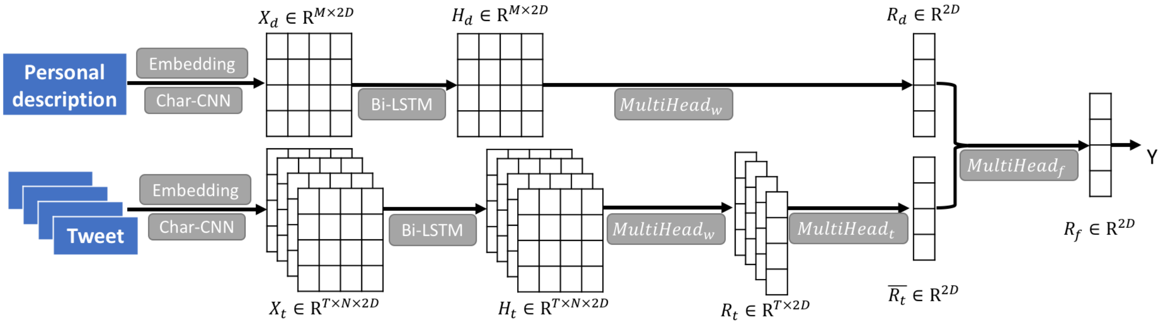


Fig. 3.1 The architecture of hierarchical self-attention neural networks for identity classification.

3.3.1 Word Embedding

Our model first maps each word in description and tweets into a word embedding space $\in R^{V\times D}$ by a table lookup operation, where V is the vocabulary size, and D is the embedding dimension.

Because of the noisy nature of tweet text, we further use a character-level convolutional neural network to generate character-level word embeddings, which are helpful for dealing with out of vocabulary tokens. More specifically, for each character c_i in a word $w = (c_1,...,c_k)$, we first map it into a character embedding space and get $v_{c_i} \in R^d$. Then a convolutional neural network is applied to generate features from characters [63]. For a character window $v_{c_i:c_{i+h-1}} \in R^{h\times d}$, a feature θ_i is generated by $\theta_i = f(w \cdot v_{c_i:c_{i+h-1}} + b)$ where $w \in R^{h\times d}$ and b are a convolution filter and a bias term respectively, $f(\cdot)$ is a non-linear function *relu*. Sliding the filter from the beginning of the character embedding matrix till the end, we get a feature vector $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_{k-h+1}]$. Then, we apply max pooling over this vector to get the most representative feature. With D such convolutional filters, we get the character-level word embedding for word w.

The final vector representation $v_w \in R^{2D}$ for word w is just the concatenation of its general word embedding vector and character-level word embedding vector. Given one description with M tokens and T tweets each with N tokens, we get two embedding matrices $X_d \in R^{M\times 2D}$ and $X_t \in R^{T\times N\times 2D}$ for description and tweets respectively.

3.3.2 Bi-LSTM

After get the embedding matrices for tweets and description, we use a bidirectional LSTM to extract context specific features from each text. At each time step, one forward LSTM takes the current word vector v_{w_i} and the previous hidden state $\overrightarrow{h_{w_{i-1}}}$ to generate the hidden state for word w_i. Another backward LSTM generates another sequence of hidden states in the reversed direction.

$$\begin{aligned}\overrightarrow{h_{w_i}} &= \overrightarrow{LSTM}(v_{w_i}, \overrightarrow{h_{w_{i-1}}})\\ \overleftarrow{h_{w_i}} &= \overleftarrow{LSTM}(v_{w_i}, \overleftarrow{h_{w_{i+1}}})\end{aligned} \tag{3.1}$$

The final hidden state $h_{w_i} \in R^{2D}$ for word w_i is the concatenation of $\overrightarrow{h_{w_i}}$ and $\overleftarrow{h_{w_i}}$ as $h_{w_i} = [\overrightarrow{h_{w_i}}, \overleftarrow{h_{w_i}}]$. With T tweets and one description, we get two hidden state matrices $H_t \in R^{T\times N\times 2D}$ and $H_d \in R^{M\times 2D}$.

3.3.3 Attention

Following the Bi-LSTM layer, we use a word-level multi-head attention layer to find important words in a text [112].

Specifically, a multi-head attention is computed as follows:

$$MultiHead(H_d) = Concat(head_1, ..., head_h)W^O$$
$$head_i = softmax(\frac{H_d W_i^Q \cdot (H_d W_i^K)^T}{\sqrt{d_k}})H_d W_i^V$$

where $d_k = 2D/h$, W_i^Q, W_i^K, $W_i^V \in R^{2D \times d_k}$, and $W^O \in R^{hd_k \times 2D}$ are projection parameters for query, key, value, and output respectively.

Take a user description for example. Given the hidden state matrix H_d of the description, each head first projects H_d into three subspaces — query $H_d W_i^Q$, key $H_d W_i^K$, and value $H_d W_i^V$. The matrix product between key and query after softmax normalization is the self-attention, which indicates important parts in the value matrix. The multiplication of self-attention and value matrix is the output of this attention head. The final output of multi-head attention is the concatenation of h such heads after projection by W^O.

After this word-level attention layer, we apply a row-wise average pooling to get a high-level representation vector for description.

$$R_d = row\_avg(MultiHead_w(H_d)) \in R^{2D} \tag{3.2}$$

Similarly, we can get T representation vectors from T tweets using the same word-level attention, which forms $R_t \in R^{T \times 2D}$.

Further, a tweet-level multi-head attention layer computes the final tweets representation vector $\bar{R_t}$ as follows:

$$\bar{R_t} = row\_avg(MultiHead_t(R_t)) \in R^{2D} \tag{3.3}$$

In practise, we also tried using an additional Bi-LSTM layer to model the sequence of tweets, but we did not observe any significant performance gain.

Given the description representation R_d and tweets representation $\bar{R_t}$, a field attention generates the final classification feature vector

$$R_f = row\_avg(MultiHead_f([R_d; \bar{R_t}])) \tag{3.4}$$

where $[R_d; \bar{R}_t] \in R^{2\times 2D}$ means concatenating by row.

3.3.4 Final Classification

Finally, the probability for each identity is computed by a softmax function:

$$P = softmax(WR_f + b) \tag{3.5}$$

where $W \in R^{|C|\times 2D}$ is the projection parameter, $b \in R^{|C|}$ is the bias term, and C is the set of identity classes. We minimize the cross-entropy loss function to train our model.

3.4 Experiments

3.4.1 Dataset

To examine the effectiveness of our method, we collect two datasets from Twitter. The first is a public figure dataset. We use Twitter's verification as a proxy for public figures. These verified accounts include users in music, government, sports, business, and etc[1]. We sampled 156746 verified accounts and 376371 unverified accounts through Twitter's sample stream data [2]. Then we collected their most recent 20 tweets from Twitter's API in November 2018. We randomly choose 5000 users as a development set and 10000 users as a test set. A summary of this dataset is shown in Table 3.1.

	Public Figure		Identity						
	Verified	Unverified	Media	Reporter	Celebrity	Government	Company	Sport	Regular
Train	152368	365749	1140	614	876	844	879	870	6623
Dev.	1452	3548	52	23	38	40	35	43	269
Test	2926	7074	97	39	75	81	66	74	568

Table 3.1 A brief summary of our two datasets.

In addition, we introduce another human labeled identity dataset for more fine-grained identity classification, which contains seven identity classes: "news media", "news reporter", "government official", "celebrity", "company", "sport", and "regular people". For each identity, we manually labelled thousands of Twitter users and collected their most recent 20 tweets for classification in November 2018. For the regular Twitter users, we randomly sampled them from the Twitter sample stream. News media accounts are these official accounts of news websites like BBC. News

[1] https://help.twitter.com/en/managing-your-account/about-twitter-verified-accounts

[2] https://developer.twitter.com/en/docs/tweets/sample-realtime/overview/GET_statuse_sample.html

News Media	News Reporter	Celebrity	Government Official	Company	Sport
CBSNews	PamelaPaulNYT	aliciakeys	USDOL	VisualStudio	NBA
earthtimes	HowardKurtz	Adele	RepRichmond	lifeatgoogle	Pirates
BBCNewsAsia	jennaportnoy	GreenDay	HouseGOP	BMW	NFL
phillydotcom	wpjenna	ladygaga	BelgiumNATO	AEO	KKRiders
TheSiasatDaily	twithersAP	TheEllenShow	usafpressdesk	Sony	USAGym

Table 3.2 Five representative Twitter handles for each identity class except for regular users.

reporters are mainly composed of news editors or journalists. Government officials represent government offices or politicians. We collected these three types of accounts from corresponding official websites. For the other three categories, we first search Twitter for these three categories, and then we downloaded their most recent tweets using Twitter's API. Two individual workers labeled these users independently, and we include users that both two workers agreed on. The inter-rater agreement measure is 0.96. In Table 3.2, we list several representative Twitter handles for each identity class except for regular users. Table 2.2 shows a summary of this dataset. Since regular users are the majority of Twitter users, about half of the users in this dataset are regular users.

This paper focuses on a content-based approach for identity classification, so we only use personal description and text of each tweet for each user.

3.4.2 Hyperparameter Setting

In our experiments, we initialize the general word embeddings with released 300-dimensional Glove vectors[3] [83]. For words not appearing in Glove vocabulary, we randomly initialize them from a uniform distribution $U(-0.25, 0.25)$. The 100-dimensional character embeddings are initialized with a uniform distribution $U(-1.0, 1.0)$. These embeddings are adapted during training. We use filter windows of size 3,4,5 with 100 feature maps each. The state dimension D of LSTM is chosen as 300. For all the multi-head attention layers, we choose the number of heads as 6. We apply dropout [105] on the input of Bi-LSTM layer and also the output of the softmax function in these attention layers. The dropout rate is chosen as 0.5. The batch size is 32. We use Adam update rule [64] to optimize our model. The initial learning rate is 10^{-4} and it drops to 10^{-5} at the last 1/3 epochs. We train our model 10 epochs, and every 100 steps we evaluate our method on development set and save the model with the best result. All these hyperparameters are tuned on the development set of identity dataset.

[3] https://nlp.stanford.edu/projects/glove/

3.4.3 Baselines

MNB: Multinomial Naive Bayes classifier with unigrams and bigrams. The term features are weighted by their TF-IDF scores. Additive smoothing parameter is set as 10^{-4} via a grid search on the development set of identity dataset.
SVM: Support Vector Machine classifier with unigrams and linear kernel. The term features are weighted by their TF-IDF scores. Penalty parameter is set as 100 via a grid search on the development set of identity dataset.
CNN: Convolutional Neural Networks [63] with filter window size 3,4,5 and 100 feature maps each. Initial learning rate is 10^{-3} and drops to 10^{-4} at the last 1/3 epochs.
Bi-LSTM: Bidirectional-LSTM model with 300 hidden states in each direction. The average of output at each step is used for the final classification.
Bi-LSTM-ATT: Bidirectional-LSTM model enhanced with self-attention. We use multi-head attention with 6 heads.
fastText [61]: we set word embedding size as 300, use unigram, and train it 10 epochs with initial learning 1.0.
For methods above, we combine personal description and tweets into a whole document for each user.

3.4.4 Results

Table 3.3 Comparisons between our methods and baselines on identity classification.

		Public Figure		Identity	
		Accuracy	Macro-F1	Accuracy	Macro-F1
Baselines	MNB	81.81	82.79	82.9	75.91
	SVM	90.60	88.59	85.9	80.19
	fastText	90.93	89.01	85.7	80.01
	CNN	91.45	89.85	85.9	81.24
	Bi-LSTM	93.10	91.84	86.5	84.25
	Bi-LSTM-ATT	93.23	91.94	87.3	83.35
Ablated Models	w/o attentions	93.78	92.45	87.0	83.26
	w/o charcnn	93.47	92.23	89.0	85.39
	w/o description	92.39	90.90	86.7	81.56
	w/o tweets	91.62	89.77	84.2	78.41
	Full Model	**94.21**	**93.07**	**89.5**	**86.09**
	Full Model-transfer			**91.6**	**88.63**

In Table 3.3, we show comparison results between our model and baselines. Generally, LSTM based methods work the best among all these baseline approaches. SVM

Table 3.4 The effectiveness of different levels of attentions tested on the identity dataset.

	Accuracy	Macro-F1
Full Model	89.5	86.09
w/o word attention	88.8	84.41
w/o field attention	88.5	85.24
w/o tweet attention	88.5	84.6
w/o all attention	87.0	83.26

has comparable performance to these neural network based methods on the identity dataset, but falls behind on the larger public figure dataset.

Our method outperforms these baselines on both datasets, especially for the more challenging fine-grained identity classification task. Our model can successfully identify public figures with accuracy 94.21% and classify identity with accuracy 89.5%. Compared to a strong baseline Bi-LSTM-ATT, our model achieves a 2.2% increase in accuracy, which shows that our model with structured input has better classification capability.

We further performed ablation studies to analyze the contribution of each model component, where we removed attention modules, character-level word embeddings, tweet texts, and user description one by one at a time. As shown in Table 4.3, attention modules make a great contribution to the final classification performance, especially for the more fine-grained task. We present the performance breakdown for each attention module in Table 3.4. Each level of attention effectively improve the performance of our model. Recognizing important words, tweets, and feature fields at different levels is helpful for learning classification representations. According to Table 4.3, the character-level convolutional layer is also helpful for capturing some character-level patterns.

We also examined the impact of two different text fields: personal description and tweets. Indeed, we found that what users tweeted about is more important than what they described themselves. On both datasets, users' tweets provide more discriminative power than users' personal descriptions.

3.4.5 Transfer Learning for Fine-grained Identity Classification

In reality, it is expensive to get a large-scale human labeled dataset for training a fine-grained identity classifier. However, a well-known drawback of neural network based methods is that they require a lot of data for training. Recently, learning from massive data and transferring learned knowledge to other tasks attracts a lot of attention [85, 33]. Since it is relatively easier to get a coarse-grained public figure dataset to classify those public figures, we explore how to use this coarse-grained public figure dataset to help the training of fine-grained identity classifier.

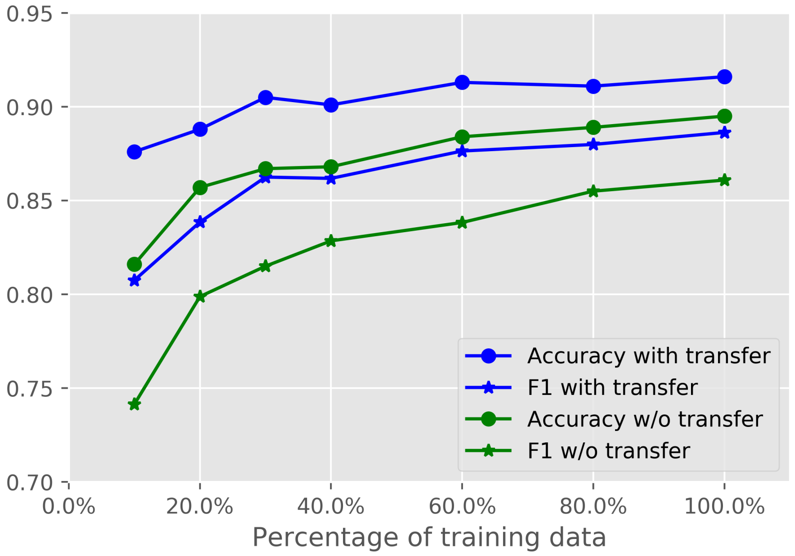


Fig. 3.2 Performance comparison between our model with transfer learning and without. We train our model on various amounts of training data.

Specifically, we first pretrain a binary classifier on the public figure dataset and save the best trained model on its development set. To make a fair comparison, we excluded all the users appearing in identity dataset from the public figure dataset when we built our datasets. Then we initialize the fine-grained identity classifier with this pretrained model except for the final classification layer. After such initialization step, we first train the final classification layer for 3 epochs with learning rate 0.01, and then train our full identity classification model with the same procedure as before. We observe a big performance boost when we apply such pretraining as shown in Table 4.3. The classification accuracy for the fine-grained task increases by 2.1% with transfer learning.

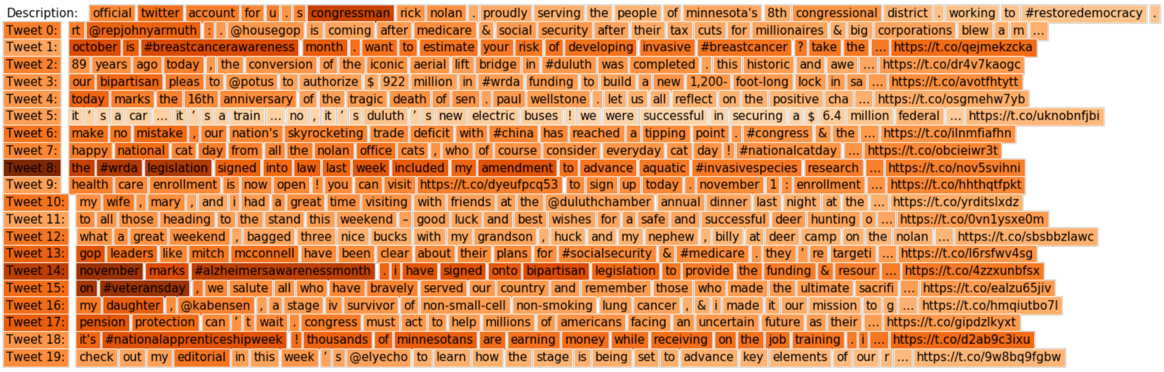
We further examined the performance of our model with pretraining using various amounts of training data. As shown in Figure 3.2, our pretrained model reaches a comparable performance only with 20%-30% labeled training data when compared to the model trained on full identity dataset without pretraining. Using only 20% of training data, we can get accuracy 0.888 and F1 0.839. If we increase the data size to 30% of the training data, the accuracy and F1 will increase to 0.905 and 0.863 respectively. Such pretraining makes great improvements over fine-grained identity classification especially when we lack labeled training data.

3.4.6 Case Study

In this section, we present a case study in the test set of identity dataset to show the effectiveness of our model. Because of the difficulties of visualizing and interpreting multi-head attention weights, we instead average over the attention weights in multiple heads which gives us an approximation of the importance of each word in texts. Take the user description for example, the approximated importance weight of each word in the description is given by

$$\alpha_d = row\_avg(\frac{1}{h}\sum_i softmax(\frac{H_d W_i^Q (H_d W_i^K)^T}{\sqrt{d_k}}))$$

Similarly, we can get the importance weights for tweets as well as words in tweets.



Description: official twitter account for u . s congressman rick nolan . proudly serving the people of minnesota's 8th congressional district . working to #restoredemocracy .

Tweet 0: rt @repjohnyarmuth : . @housegop is coming after medicare & social security after their tax cuts for millionaires & big corporations blew a m ...

Tweet 1: october is #breastcancerawareness month . want to estimate your risk of developing invasive #breastcancer ? take the ... https://t.co/qejmekzcka

Tweet 2: 89 years ago today , the conversion of the iconic aerial lift bridge in #duluth was completed . this historic and awe ... https://t.co/dr4v7kaogc

Tweet 3: our bipartisan pleas to @potus to authorize $ 922 million in #wrda funding to build a new 1,200- foot-long lock in sa ... https://t.co/avotfhtytt

Tweet 4: today marks the 16th anniversary of the tragic death of sen . paul wellstone . let us all reflect on the positive cha ... https://t.co/osgmehw7yb

Tweet 5: it ’ s a car ... it ’ s a train ... no , it ’ s duluth ’ s new electric buses ! we were successful in securing a $ 6.4 million federal ... https://t.co/uknobnfjbi

Tweet 6: make no mistake , our nation's skyrocketing trade deficit with #china has reached a tipping point . #congress & the ... https://t.co/ilnmfiafhn

Tweet 7: happy national cat day from all the nolan office cats , who of course consider everyday cat day ! #nationalcatday ... https://t.co/obcieiwr3t

Tweet 8: the #wrda legislation signed into law last week included my amendment to advance aquatic #invasivespecies research ... https://t.co/nov5svihni

Tweet 9: health care enrollment is now open ! you can visit https://t.co/dyeufpcq53 to sign up today . november 1 : enrollment ... https://t.co/hhthqtfpkt

Tweet 10: my wife , mary , and i had a great time visiting with friends at the @duluthchamber annual dinner last night at the ... https://t.co/yrditslxdz

Tweet 11: to all those heading to the stand this weekend – good luck and best wishes for a safe and successful deer hunting o ... https://t.co/0vn1ysxe0m

Tweet 12: what a great weekend , bagged three nice bucks with my grandson , huck and my nephew , billy at deer camp on the nolan ... https://t.co/sbsbbzlawc

Tweet 13: gop leaders like mitch mcconnell have been clear about their plans for #socialsecurity & #medicare . they ’ re targeti ... https://t.co/l6rsfwv4sg

Tweet 14: november marks #alzheimersawarenessmonth . i have signed onto bipartisan legislation to provide the funding & resour ... https://t.co/4zzxunbfsx

Tweet 15: on #veteransday , we salute all who have bravely served our country and remember those who made the ultimate sacrifi ... https://t.co/ealzu65jiv

Tweet 16: my daughter , @kabensen , a stage iv survivor of non-small-cell non-smoking lung cancer , & i made it our mission to g ... https://t.co/hmqiutbo7l

Tweet 17: pension protection can ’ t wait . congress must act to help millions of americans facing an uncertain future as their ... https://t.co/gipdzlkyxt

Tweet 18: it's #nationalapprenticeshipweek ! thousands of minnesotans are earning money while receiving on the job training . i ... https://t.co/d2ab9c3ixu

Tweet 19: check out my editorial in this week ’ s @elyecho to learn how the stage is being set to advance key elements of our r ... https://t.co/9w8bq9fgbw

Fig. 3.3 The visualization of attention weights for each tweet and description. The color depth denotes the importance degree of a word per tweet. The importance of each tweet is depicted as the background color of corresponding tweet header.

In Figure 3.3, we show twenty tweets and a description from a government official user. We use the background color to represent importance weight for each word. The color depth denotes the importance degree of a word per tweet. We plot the tweet-level importance weights as the background color of tweet index at the beginning of each tweet. As shown in this figure, words like “congressman”, “legislation” in this user’s description are important clues indicating his/her identity. From the tweet-level attention, we know that 8th and 14th tweets are the most important tweets related with the identity because they include words like “legislation” and “bipartisan”. On

Prediction Truth	Regular	Media	Celebrity	Sport	Company	Government	Reporter
Regular	535	10	12	0	5	2	4
Media	6	81	1	4	2	2	1
Celebrity	15	0	55	2	2	1	0
Sport	1	1	0	71	1	0	0
Company	1	2	1	4	58	0	0
Government	1	1	0	0	0	79	0
Reporter	1	0	1	0	0	0	37

Table 3.5 The confusion matrix on the test set of identity dataset

the contrary, 5th tweet of this user only contain some general words like "car", which makes it less important than other tweets.

3.4.7 Error Analysis

We perform an error analysis to investigate why our model fails in certain cases. Table 3.5 shows the confusion matrix generated from prediction results of our identity dataset. As shown in this table, it is relatively harder for our model to distinguish between celebrities and regular users. We further looked at such errors with high confidences and found that some celebrities just have not posted any indicating words in their tweets or descriptions. For example, one celebrity account only use "A Virgo" in the description without any other words, which makes this account predicted as a regular user. Including other features like number of followers or network connections may overcome this issue, and we leave it for future work. Another common error happens when dealing with non-English tweets. Even enhanced with transferred knowledge from the large-scale verify dataset, our model still cannot handle some rare languages in the data.

In Table 3.6, we also show the testing performance for users with different languages. Since most of our labeled users speak English, the identity classifier works better in this case. For other users who use languages such as Spanish and Portuguese, the performance is much worse especially for the F1 score. For future work, we should take more labeled non-English speakers into consideration.

3.5 Discussion & Conclusion

As previously discussed, identities can vary in granularity. We examined two levels - coarse grained (verified or not) and more fine grained (news media, government officials,

lang	# of users	Accuracy	Macro-F1
English	574	91.11	90.35
Spanish	94	88.30	70.51
Arabic	51	96.08	96.64
Portuguese	36	91.67	65.10
French	20	90.0	55.26

Table 3.6 Testing performance for users with different languages (top 5).

etc.). However, there could be more levels. This limits our understanding of activities of online actors with those identities. A hierarchical approach for identity classification might be worth further research. Future research should take this into consideration and learn users' identities in a more flexible way. Besides, because of the nature of social media, the content on Twitter would evolve rapidly. In order to deploy our method in real-time, we need consider an online learning procedure that adapts our model to new data patterns. Since our method is purely content-based, potential improvements could be made using additional information like the number of users' followers, users' network connections, and even their profile images. We leave this as our future work.

In the real-world people often have multiple identities - e.g., Serbian, Entrepreneur, Policewoman, Woman, Mother. The question is what is the relation between identities, users, and user accounts. Herein, we treat each account as a different user. However, in social media, some people use different accounts and/or different social media platforms for different identities - e.g., Facebook for Mother, Twitter for Entrepreneur and a separate Twitter handle for official policewoman account. In this paper, we made no effort to determine whether an individual had multiple accounts. Thus, the same user may get multiple classifications if that user has multiple accounts. Future work should explore how to link multiple identities to the same user. To this point, when there is either a hierarchy of identities or orthogonal identity categories, then using identities at different levels of granularity, as we did herein, enables multiple identities to be assigned to the same account and so to the same user.

In conclusion, we introduce two datasets for online user identity classification. One is automatically extracted from Twitter, the other is a manually labelled dataset. We present a novel content-based method for classifying social media users into a set of identities (social roles) on Twitter. Our experiments on two datasets show that our model significantly outperforms multiple baseline approaches. Using one personal description and up to twenty tweets for each user, we can identify public figures with accuracy 94.21% and classify more fine-grained identities with accuracy 89.5%. We proposed and tested a transfer learning scheme that further boosts the final identity classification accuracy by a large margin. Though, the focus of this paper is learning users' social identities. It is possible to extend this work to predict other demographics like gender and age.

Chapter 4

Hierarchical User Location Prediction

4.1 Introduction

Accurate estimation of user location is an important factor for many online services, such as recommendation systems [89], event detection [100], and disaster management [22]. Though internet service providers can directly obtain users' location information from some explicit metadata like IP address and GPS signal, such private information is not available for third-party contributors. With this motivation, researchers have developed location prediction systems for various platforms, such as Wikipedia [81], Facebook [9], and Twitter [45].

In the case of Twitter, due to the sparsity of geotagged tweets [40] and the unreliability of user self-declared home location in profile [50], there is a growing body of research trying to determine users' locations automatically. Various methods have been proposed for this purpose. They can be roughly divided into three categories. The first type consists of tweet text-based methods, where the word distribution is used to estimate geolocations of users [98, 116]. In the second type, methods combining metadata features such as time zone, profile description are developed to improve performance [46]. Network-based methods form the last type. Several studies have shown that incorporating friends' information is very useful for this task [78, 37]. Empirically, models enhanced with network information works better than the other two types, but they do not scale well to larger datasets [91].

In recent years, neural network based prediction methods have shown great success on this Twitter user geolocation prediction task [92, 78]. However, these neural network based methods largely ignore the hierarchical structure among locations (eg. country versus city), which have been shown very useful in previous study [72, 115]. In recent work, Huang and Carley [54] also demonstrate that country-level location prediction is much easier than city-level location prediction. It is natural to ask whether we can incorporate the hierarchical structure among locations into a neural network and use the coarse-grained location prediction to guide the fine-grained prediction.

In this paper, we present a hierarchical location prediction neural network (HLPNN) for user geolocation on Twitter. Our model combines text features, metadata features (personal description, profile location, name, user language, time zone), and network features together, and learns two classification representations for country-level and city-level predictions respectively. It first computes the country-level prediction, which is further used to guide the city-level prediction. Our model is flexible in accommodating different feature combinations, and it achieves state-of-the-art results under various feature settings.

4.2 Related Work

As a popular user profiling task, location inference has been widely studied in the literature. Though there are some potential privacy concerns, accurate user geolocation is a key factor for many important applications such as earthquake detection [36], and disaster management [22]. Because of insufficient geotagged data on Twitter [40], there is a growing interest in predicting Twitter users' locations.

Early work tried to identify users' locations by mapping their IP addresses to physical locations [19]. However, such private information is only accessible to internet service providers. There is no easy way for a third-party to find Twitter users' IP addresses. Later, various text-based location prediction systems were proposed. Bilhaut et al. [15] utilize a geographical gazetteer as an external lexicon and present a rule-based geographical references recognizer. Amitay et al. [4] extracted location-related information listed in a gazetteer from web content to identify geographical regions of webpages. However, as shown in [10], performances of gazetteer-based methods are hindered by the noisy and informal nature of tweets.

Moving beyond methods replying on external knowledge sources (eg. IP and gazetteers), many machine learning based methods have recently been applied to location prediction. Typically, researchers first represent locations as earth grids [116, 98], regions [79], or cities [46]. Then location classifiers are built to categorize users into different locations. Han et al. [45] first utilized feature selection methods to find location indicative words, then they used multinomial naive Bayes and logistic regression classifiers to find correct locations. Han et al. [46] further present a stacking based method that combines tweet text and metadata together. Along with these classification methods, some approaches also try to learn topic regions automatically by topic modeling, but these do not scale well to the magnitude of social media [53, 118].

Recently, deep neural network based methods are becoming popular for location prediction [77]. Huang and Carley [54] integrate text and user profile metadata into a single model using convolutional neural networks, and their experiments show superior performance over stacked naive Bayes classifiers. Miura et al. [78], Ebrahimi et al. [37] incorporate user network connection information into their neural models, where they use network embeddings to represent users in a social network. Rahimi et al. [93] also uses text and network feature together, but their approach is based on graph convolutional neural networks.

Similar to our method, some research has tried to predict user location hierarchically [72, 115]. Mahmud et al. [72] develop a two-level hierarchical location classifier which first predicts a coarse-grained location (country, time zone), and then predicts the city label within the corresponding coarse region. Wing and Baldridge [115] build a hierarchical tree of earth grids. The probability of a final fine-grained location can be computed recursively from the root node to the leaf node. Both methods have to train one classifier separately for each parent node, which is quite time-consuming for training deep neural network based methods. Additionally, certain coarse-grained locations may not have enough data samples to train a local neural classifier alone. Our hierarchical location prediction neural network overcomes these issues and only needs to be trained once.

4.3 Method

There are seven features we want to utilize in our model — tweet text, personal description, profile location, name, user language, time zone, and mention network. The first four features are text fields where users can write anything they want. User language and time zone are two categorical features that are selected by users in their profiles. The mention network is constructed directly from mentions in tweets.

We propose a hierarchical prediction framework that combines all seven features together for user location prediction. Our model first predicts the home country of a Twitter user, then uses the country-level prediction to guide the city-level prediction.

The overall architecture of our hierarchical location prediction model is shown in Figure 4.1. It first maps four text features into a word embedding space. A bidirectional

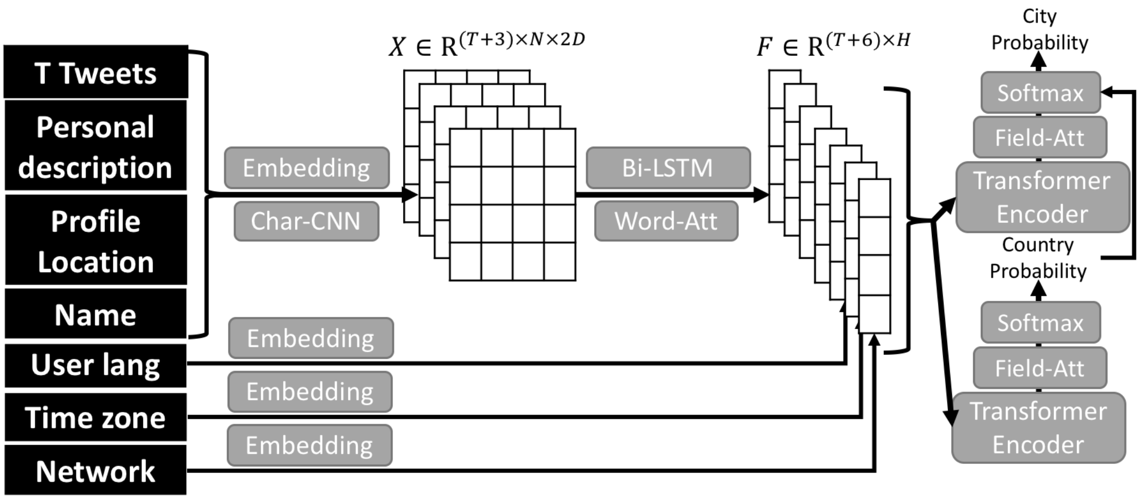


Fig. 4.1 The architecture of our hierarchical location prediction neural network.

LSTM (Bi-LSTM) neural network [52] is used to extract location-specific features from these text embedding vectors. Following Bi-LSTM, we use a word-level attention layer to generate representation vectors for these text fields. Combining all the text representations, a user language embedding, a timezone embedding, and a network embedding, we apply several layers of transformer encoders [112] to learn the correlation among all the feature fields. The probability for each country is computed after a field-level attention layer. Finally, we use the country probability as a constraint for the city-level location prediction. We elaborate details of our model in following sections.

4.3.1 Word Embedding

Assume one user has T tweets, there are $T+3$ text fields for this user including personal description, profile location, and name. We first map each word in these $T+3$ text fields into a low dimensional embedding space. The embedding vector for word w is computed as $x_w = [E(w), CNN_c(w)]$, where [,] denotes vector concatenation. $E(w)$ is the word-level embedding retrieved directly from an Embedding matrix $E \in R^{V\times D}$ by a lookup operation, where V is the vocabulary size, and D is the word-level embedding dimension. $CNN_c(w)$ is a character-level word embedding that is generated from a character-level convolutional layer. Using character-level word embeddings is helpful for dealing with out-of-vocabulary tokens and overcoming the noisy nature of tweet text.

The character-level word embedding generation process is as follows. For a character c_i in the word $w = (c_1, ..., c_k)$, we map it into a character embedding space and get a vector $v_{c_i} \in R^d$. In the convolutional layer, each filter $u \in R^{l_c \times d}$ generates a feature vector $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_{k-l_c+1}] \in R^{k-l_c+1}$, where $\theta_i = relu(u \circ v_{c_i:c_{i+l_c-1}} + b)$. b is a bias term, and "$\circ$" denotes element-wise inner product between u and character window $v_{c_i:c_{i+l_c-1}} \in R^{l_c \times d}$. After this convolutional operation, we use a max-pooling operation to select the most representative feature $\hat{\theta} = max(\boldsymbol{\theta})$. With D such filters, we get the character-level word embedding $CNN_c(w) \in R^D$.

4.3.2 Text Representation

After the word embedding layer, every word in these $T+3$ texts are transformed into a $2D$ dimension vector. Given a text with word sequence $(w_1, ..., w_N)$, we get a word embedding matrix $X \in R^{N \times 2D}$ from the embedding layer. We then apply a Bi-LSTM neural network to extract high-level semantic representations from text embedding matrices.

At every time step i, a forward LSTM takes the word embedding X_i of word w_i and previous state $\overrightarrow{h_{i-1}}$ as inputs, and generates the current hidden state $\overrightarrow{h}_i$. A backward LSTM reads the text from w_N to w_1 and generates another state sequence. The hidden state $h_i \in R^{2D}$ for word w_i is the concatenation of $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$. Concatenating all the hidden states, we get a semantic matrix $H \in R^{N \times 2D}$

$$\begin{aligned} \overrightarrow{h_i} &= \overrightarrow{LSTM}(X_i, \overrightarrow{h_{i-1}}) \\ \overleftarrow{h_i} &= \overleftarrow{LSTM}(X_i, \overleftarrow{h_{i+1}}) \\ h_i &= [\overrightarrow{h_i}, \overleftarrow{h_i}] \end{aligned} \tag{4.1}$$

Because not all words in a text contribute equally towards location prediction, we further use a multi-head attention layer [112] to generate a representation vector $f \in R^{2D}$ for each text. There are h attention heads that allow the model to attend to important information from different representation subspaces. Each head computes a text representation as a weighted average of these word hidden states. The computation steps in a multi-head attention layer are as follows.

$$\begin{aligned} f = MultiHead(q, H) &= [head_1, ..., head_h]W^O \\ head_i(q, H) &= softmax(\frac{qW_i^Q \cdot (HW_i^K)^T}{\sqrt{d_k}})HW_i^V \end{aligned}$$

where $q \in R^{2d}$ is an attention context vector learned during training, $W_i^Q, W_i^K, W_i^V \in R^{2D \times d_k}$, and $W^O \in R^{2D \times 2D}$ are projection parameters, $d_k = 2D/h$. An attention head $head_i$ first projects the attention context q and the semantic matrix H into query and key subspaces by W_i^Q, W_i^K respectively. The matrix product between query qW_i^Q and key HW_i^K after softmax normalization is an attention weight that indicates important words among the projected value vectors HW_i^V. Concatenating h heads together, we get one representation vector $f \in R^{2D}$ after projection by W^O for each text field.

4.3.3 Feature Fusion

For two categorical features, we assign an embedding vector with dimension $2D$ for each time zone and language. These embedding vectors are learned during training. We pretrain network embeddings for users involved in the mention network using LINE [109]. Network embeddings are fixed during training. We get a feature matrix $F \in R^{(T+6)\times 2D}$ by concatenating text representations of $T+3$ text fields, two embedding vectors of categorical features, and one network embedding vector.

We further use several layers of transformer encoders [112] to learn the correlation between different feature fields. Each layer consists of a multi-head self-attention network and a feed-forward network (FFN). One transformer encoder layer first uses input feature to attend important information in the feature itself by a multi-head attention sub-layer. Then a linear transformation sub-layer FFN is applied to each position identically. Same as [112], we employ residual connection [49] and layer normalization [6] around each of the two sub-layers. The output F_1 of the first transformer encoder layer is generated as follows.

$$\begin{aligned} F' &= LayerNorm(MultiHead(F,F)+F) \\ F_1 &= LayerNorm(FFN(F')+F') \end{aligned}$$

where $FFN(F') = max(0, F'W_1 + b_1)W_2 + b2$, $W_1 \in R^{2D\times D_{ff}}$, and $W_2 \in R^{D_{ff}\times 2D}$.

Since there is no position information in the transformer encoder layer, our model cannot distinguish between different types of features, eg. tweet text and personal description. To overcome this issue, we add feature type embeddings to the input representations F. There are seven features in total. Each of them has a learned feature type embedding with dimension $2D$ so that one feature type embedding and the representation of the corresponding feature can be summed.

Because the input and the output of transformer encoder have the same dimension, we stack L layers of transformer encoders to learn representations for country-level prediction and city-level prediction respectively. These two sets of encoders share the same input F, but generate different representations F_{co}^L and F_{ci}^L for country and city predictions.

The final classification features for country-level and city-level location predictions are the row-wise weighted average of F_{co} and F_{ci}. Similar to the word-level attention, we use a field-level multi-head attention layer to select important features from $T+6$ vectors and fuse them into a single vector.

$$F_{co} = MultiHead(q_{co}, F^L_{co})$$
$$F_{ci} = MultiHead(q_{ci}, F^L_{ci})$$

where $q_{co}, q_{ci} \in R^{2D}$ are two attention context vectors.

4.3.4 Hierarchical Location Prediction

The final probability for each country is computed by a softmax function

$$P_{co} = softmax(W_{co}F_{co} + b_{co})$$

where $W_{co} \in R^{M_{co} \times 2D}$ is a linear projection parameter, $b_{co} \in R^{M_{co}}$ is a bias term, and M_{co} is the number of countries.

After we get the probability for each country, we further use it to constrain the city-level prediction

$$P_{ci} = softmax(W_{ci}F_{ci} + b_{ci} + \lambda P_{co}Bias)$$

where $W_{ci} \in R^{M_{ci} \times 2D}$ is a linear projection parameter, $b_{ci} \in R^{M_{ci}}$ is a bias term, and M_{ci} is the number of cities. $Bias \in R^{M_{co} \times M_{ci}}$ is the country-city correlation matrix. If city j belongs to country i, then $Bias_{ij}$ is 0, otherwise -1. λ is a penalty term learned during training. The larger of λ, the stronger of the country constraint. In practise, we also experimented with letting the model learn the country-city correlation matrix during training, which yields similar performance.

We minimize the sum of two cross-entropy losses for country-level prediction and city-level prediction.

$$loss = -(Y_{ci} \cdot logP_{ci} + \alpha Y_{co} \cdot logP_{co})$$

where Y_{ci} and Y_{co} are one-hot encodings of city and country labels. α is the weight to control the importance of country-level supervision signal. Since a large α would potentially interfere with the training process of city-level prediction, we just set it as 1 in our experiments. Tuning this parameter on each dataset may further improve the performance.

4.4 Experiment Settings

4.4.1 Datasets

To validate our method, we use three widely adopted Twitter location prediction datasets. Table 4.1 shows a brief summary of these three datasets. They are listed as follows.

Twitter-US is a dataset compiled by Roller et al. [98]. It contains 429K training users, 10K development users, and 10K test users in North America. The ground truth location of each user is set to the first geotag of this user in the dataset. We assign the closest city to each user's ground truth location using the city category built by Han et al. [45]. Since this dataset only covers North America, we change the first level location prediction from countries to administrative regions (eg. state or province). The administrative region for each city is obtained from the original city category.

Twitter-World is a Twitter dataset covering the whole world, with 1,367K training users, 10K development users, and 10K test users [45]. The ground truth location for each user is the center of the closest city to the first geotag of this user. Only English tweets are included in this dataset, which makes it more challenging for a global-level location prediction task.

We downloaded these two datasets from Github [1]. Each user in these two datasets is represented by the concatenation of their tweets, followed by the geo-coordinates. We queried Twitter's API to add user metadata information to these two datasets in February 2019. We only get metadata for about 53% and 67% users in Twitter-US and Twitter-World respectively. Because of Twitter's privacy policy change, we could not get the time zone information anymore at the time of collection.

WNUT was released in the 2nd Workshop on Noisy User-generated Text [48]. The original user-level dataset consists of 1 million training users, 10K users in development set and test set each. Each user is assigned with the closest city center as the ground truth label. Because of Twitter's data sharing policy, only tweet IDs of training and development data are provided. We have to query Twitter's API to reconstruct the training and development dataset. We finished our data collection around August 2017. About 25% training and development users' data cannot be accessed at that time. The full anonymized test data is downloaded from the workshop website [2].

[1] https://github.com/afshinrahimi/geomdn

[2] https://noisy-text.github.io/2016/geo-shared-task.html

	Twitter-US			Twitter-World			WNUT		
	Train	Dev.	Test	Train	Dev.	Test	Train	Dev.	Test
# users	429K	10K	10K	1.37M	10K	10K	742K	7.46K	10K
# users with meta	228K	5.32K	5.34K	917K	6.50K	6.48K	742K	7.46K	10K
# tweets	36.4M	861K	831K	11.2M	488K	315K	8.97M	90.3K	99.7K
# tweets per user	84.60	86.14	83.12	8.16	48.83	31.59	12.09	12.10	9.97

Table 4.1 A brief summary of our datasets. For each dataset, we report the number of users, number of users with metadata, number of tweets, and average number of tweets per user. We collected metadata for 53% and 67% of users in Twitter-US and Twitter-World. Time zone information was not available when we collected metadata for these two datasets. About 25% of training and development users' data was inaccessible when we collected WNUT in 2017.

4.4.2 Text Preprocessing & Network Construction

For all the text fields, we first convert them into lower case, then use a tweet-specific tokenizer from NLTK[3] to tokenize them. To keep a reasonable vocabulary size, we only keep tokens with frequencies greater than 10 times in our word vocabulary. Our character vocabulary includes characters that appear more than 5 times in the training corpus.

We construct user networks from mentions in tweets. For WNUT, we keep users satisfying one of the following conditions in the mention network: (1) users in the original dataset (2) users who are mentioned by two different users in the dataset. For Twitter-US and Twitter-World, following previous work [93], a uni-directional edge is set if two users in our dataset directly mentioned each other, or they co-mentioned another user. We remove celebrities who are mentioned by more than 10 different users from the mentioning network. These celebrities are still kept in the dataset and their network embeddings are set as 0.

4.4.3 Evaluation Metrics

We evaluate our method using four commonly used metrics listed below.

Accuracy: The percentage of correctly predicted home cities.

Acc@161: The percentage of predicted cities which are within a 161 km (100 miles) radius of true locations to capture near-misses.

Median: The median distance measured in kilometer from the predicted city to the

[3] https://www.nltk.org/api/nltk.tokenize.html

true location coordinates.
Mean: The mean value of error distances in predictions.

4.4.4 Hyperparameter Settings

In our experiments, we initialize word embeddings with released 300-dimensional Glove vectors [83]. For words not appearing in Glove vocabulary, we randomly initialize them from a uniform distribution U(-0.25, 0.25). We choose the character embedding dimension as 50. The character embeddings are randomly initialized from a uniform distribution U(-1.0,1.0), as well as the timezone embeddings and language embeddings. These embeddings are all learned during training. Because our three datasets are sufficiently large to train our model, the learning is quite stable and performance does not fluctuate a lot.

Network embeddings are trained using LINE [109] with parameters of dimension 600, initial learning rate 0.025, order 2, negative sample size 5, and training sample size 10000M. Network embeddings are fixed during training. For users not appearing in the mention network, we set their network embedding vectors as 0.

	Twitter-US	Twitter-World	WNUT
Batch size	32	64	64
Initial learning rate	10^{-4}	10^{-4}	10^{-4}
D: Word embedding dimension	300	300	300
d: Char. embedding dimension	50	50	50
l_c: filter sizes in Char. CNN	3,4,5	3,4,5	3,4,5
Filter number for each size	100	100	100
h: number of heads	10	10	10
L: layers of transformer encoder	3	3	3
λ: initial penalty term	1	1	1
α: weight for country supervision	1	1	1
D_{ff}: inner dimension of FFN	2400	2400	2400
Max number of tweets per user	100	50	20

Table 4.2 A summary of hyperparameter settings of our model.

A brief summary of hyperparameter settings of our model is shown in Table 4.2. The initial learning rate is 10^{-4}. If the validation accuracy on the development set does not increase, we decrease the learning rate to 10^{-5} and train the model for additional

3 epochs. Empirically, training terminates within 10 epochs. Penalty λ is initialized as 1.0 and is adapted during training. We apply dropout on the input of Bi-LSTM layer and the output of two sub-layers in transformer encoders with dropout rate 0.3 and 0.1 respectively. We use the Adam update rule [64] to optimize our model. Gradients are clipped between -1 and 1. The maximum numbers of tweets per user for training and evaluating on Twitter-US are 100 and 200 respectively. We only tuned our model, learning rate, and dropout rate on the development set of WNUT.

4.5 Results

4.5.1 Baseline Comparisons

In our experiments, we evaluate our model under four different feature settings: Text, Text+Meta, Text+Network, Text+Meta+Network. HLPNN-Text is our model only using tweet text as input. HLPNN-Meta is the model that combines text and metadata (description, location, name, user language, time zone). HLPNN-Net is the model that combines text and mention network. HLPNN is our full model that uses text, metadata, and mention network for Twitter user geolocation.

We present comparisons between our model and previous work in Table 4.3. As shown in the table, our model outperforms these baselines across three datasets under various feature settings.

Only using text feature from tweets, our model HLPNN-Text works the best among all these text-based location prediction systems and wins by a large margin. It not only improves prediction accuracy but also greatly reduces mean error distance. Compared with a strong neural model equipped with local dialects [92], it increases Acc@161 by an absolute value 4% and reduces mean error distance by about 400 kilometers on the challenging Twitter-World dataset, without using any external knowledge. Its mean error distance on Twitter-World is even comparable to some methods using network feature [34].

With text and metadata, HLPNN-Meta correctly predicts locations of 57.2% users in WNUT dataset, which is even better than these location prediction systems that use text, metadata, and network. Because in the WNUT dataset the ground truth location is the closest city's center, Our model achieves 0 median error when its accuracy is greater than 50%. Note that Miura et al. [78] used 279K users added with metadata in their experiments on Twitter-US, while we use all 449K users for training and

	Twitter-US			Twitter-World			WNUT			
	Acc@161↑	Median↓	Mean↓	Acc@161↑	Median↓	Mean↓	Accuracy↑	Acc@161↑	Median↓	Mean↓
Text										
Wing and Baldridge [115]	49.2	170.5	703.6	32.7	490.0	1714.6	-	-	-	-
Rahimi et al. [94]*	50	159	686	32	530	1724	-	-	-	-
Miura et al. [78]-TEXT	55.6	110.5	585.1	-	-	-	35.4	50.3	155.8	1592.6
Rahimi et al. [92]	55	91	581	36	373	1417	-	-	-	-
HLPNN-Text	**57.1**	**89.92**	**516.6**	**40.1**	**299.1**	**1048.1**	**37.3**	**52.9**	**109.3**	**1289.4**
Text+Meta										
Miura et al. [78]-META	**67.2**	**46.8**	**356.3**	-	-	-	54.7	70.2	0	825.8
HLPNN-Meta	61.1	64.3	454.8	**56.4**	**86.2**	**762.1**	**57.2**	**73.1**	**0**	**572.5**
Text+Net										
Rahimi et al. [91]*	60	78	529	53	111	1403	-	-	-	-
Rahimi et al. [92]	61	77	515	53	104	1280	-	-	-	-
Miura et al. [78]-UNET	61.5	65	481.5	-	-	-	**38.1**	**53.3**	**99.9**	1498.6
Do et al. [34]	66.2	45	433	53.3	118	1044	-	-	-	-
Rahimi et al. [93]-MLP-TXT+NET	66	56	420	58	**53**	1030	-	-	-	-
Rahimi et al. [93]-GCN	62	71	485	54	108	1130	-	-	-	-
HLPNN-Net	**70.8**	**31.6**	**361.5**	**58.9**	59.9	**827.6**	37.8	**53.3**	105.26	**1297.7**
Text+Meta+Net										
Miura et al. [77]	-	-	-	-	-	-	47.6	-	16.1	1122.3
Jayasinghe et al. [59]	-	-	-	-	-	-	52.6	-	21.7	1928.8
Miura et al. [78]	70.1	41.9	335.7	-	-	-	56.4	71.9	**0**	780.5
HLPNN	**72.7**	**28.2**	**323.1**	**68.4**	**6.20**	**610.0**	**57.6**	**73.4**	**0**	**538.8**

Table 4.3 Comparisons between our method and baselines. We report results under four different feature settings: Text, Text+Metadata, Text+Network, Text+Metadata+Network. "-" signifies that no results were published for the given dataset, "*" denotes that results are cited from Rahimi et al. [92]. Note that Miura et al. [78] only used 279K users added with metadata in their experiments of Twitter-US.

evaluation, and only 53% of them have metadata, which makes it difficult to make a fair comparison.

Adding network feature further improves our model's performances. It achieves state-of-the-art results combining all features on these three datasets. Even though unifying network information is not the focus of this paper, our model still outperforms or has comparable results to some well-designed network-based location prediction systems like [93]. On Twitter-US dataset, our model variant HLPNN-Net achieves a 4.6% increase in Acc@161 against previous state-of-the-art methods [34] and [93]. The prediction accuracy of HLPNN-Net on WNUT dataset is similar to [78], but with a noticeable lower mean error distance.

4.5.2 Ablation Study

In this section, we provide an ablation study to examine the contribution of each model component. Specifically, we remove the character-level word embedding, the word-level attention, the field-level attention, the transformer encoders, and the country supervision signal one by one at a time. We run experiments on the WNUT dataset with text features. We also tried to run a two-step training procedure, where we train a country-level classifier first and then train city-level classifiers for each country.

	Accuracy	Acc@161	Median	Mean
HLPNN	37.3	52.9	109.3	1289.4
w/o Char-CNN	36.3	51.0	130.8	1429.9
w/o Word-Att	36.4	51.5	130.2	1377.5
w/o Field-Att	37.0	52.0	121.8	1337.5
w/o encoders	36.8	52.5	117.4	1402.9
w/o country	36.7	52.6	124.8	1399.2
Two-step training	36.3	52.2	122.1	1381.6

Table 4.4 An ablation study on WNUT dataset.

The performance breakdown for each model component is shown in Table 4.4. Compared to the full model, we can find that the character-level word embedding layer is especially helpful for dealing with noisy social media text. The word-level attention also provides performance gain, while the field-level attention only provides a marginal improvement. The reason could be the multi-head attention layers in the transformer encoders already captures important information among different feature fields. These two transformer encoders learn the correlation between features and decouple these two level predictions. Finally, using the country supervision can help model to achieve a better performance with a lower mean error distance. Two-step training does not provide better performance possible reason is lack of sufficient training data for city-level neural classifiers of each country.

4.5.3 Country Effect

To directly measure the effect of adding country-level supervision, we define a relative country error which is the percentage of city-level predictions located in incorrect countries among all misclassified city-level predictions.

$$\text{relative country error} = \frac{\text{\# of incorrect country}}{\text{\# of incorrect city}}$$

The lower this metric means the better one model can predict the city-level location, at least in the correct country.

We vary the weight α of country-level supervision signal in our loss function from 0 to 20. The larger α means the more important the country-level supervision during the optimization. When α equals 0, there is no country-level supervision in our model. As shown in Figure 4.2, increasing α would improve the relative country error from 26.2% to 23.1%, which shows the country-level supervision signal indeed can help our model predict the city-level location towards the correct country. This possibly explains why our model has a lower mean error distance when compared to other methods.

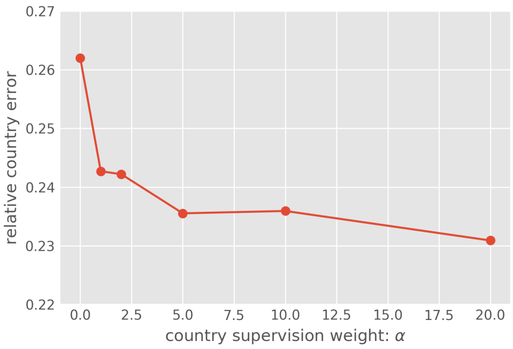


Fig. 4.2 Relative country error with varying α on test dataset. Experiments were conducted on WNUT dataset with text feature.

4.6 Conclusion

In this chapter, we propose a hierarchical location prediction neural network, which combines text, metadata, network information for user location prediction. Our model can accommodate various feature combinations. Extensive experiments have been conducted to validate the effectiveness of our model under four different feature settings across three commonly used benchmarks. Our experiments show our HLPNN model achieves state-of-the-art results on these three datasets. It not only improves the prediction accuracy but also significantly reduces the mean error distance. In our ablation analysis, we show that using character-aware word embeddings is helpful for overcoming noise in social media text. The transformer encoders effectively learn the correlation between different features and decouple the two different level predictions. In our experiments, we also analyzed the effect of adding country-level regularization. The country-level supervision could effectively guide the city-level prediction towards the correct country, and reduce the errors where users are misplaced in the wrong countries.

Though our HLPNN model achieves great performances under Text+Net and Text+Meta+Net settings, potential improvements could be made using better graph-level classification frameworks. We currently only use network information to train network embeddings as user-level features. For future work, we would like to explore ways to combine graph-level classification methods and our user-level learning model. Propagating features from connected friends would provide much more information than just using network embedding vectors. Besides, our model assumes each post of one user all comes from one single home location but ignores the dynamic user movement pattern like traveling. We plan to incorporate temporal states to capture location changes in future work.

Chapter 5

Graph-level Attributes Prediction

5.1 Introduction

In previous chapters, I have shown various methods for user attributes prediction. Most of them only use user's local features without any network information. The only exception is the hierarchical location prediction neural network where we use network embedding to provide network structure information to the model. However, a drawback of these type of embedding methods is they cannot handle newly incoming users. Test users should be available during training so that we can build a network to train embeddings like LINE [109]. Besides, such a model does not utilize features from neighbourhood friends, which may contain useful information for prediction.

In this chapter, I will explore using graph neural networks (GNN) to incorporate graph connections for training. Graph neural networks are deep learning-based methods that operate on graphs. At each layer, GNNs aggregate information from neighbourhoods and generate hidden states for each node. Because GNNs do not require a fixed graph, we can easily apply them to new graphs on the fly.

Most of the previous work either focus on classification only using network information [84], or combining network information with processed features [44]. However, in this work, users in a network are associated with raw text features, which cannot be utilized directly in a graph neural network. In this chapter, I will explore how to combine my previous user-level learning architecture with graph neural networks.

There are two main components in my proposed method. One is user-level feature extractor, which is just my previous user-level learning module. Another is a graph neural network that propagate features from users' neighbourhoods. Generally, the learning process can be written as

$$x_i = F(tweets_i, description_i) \tag{5.1}$$

$$h_i = GNN(x_i, x_{[nei_i]}) \tag{5.2}$$

$$y_i = MLP(x_i, h_i) \tag{5.3}$$

where F represents a user-level feature learning function, $x_{[nei_i]}$ are the features of neighbour users of user i. $MLP(\cdot)$ is a multilayer perception network.

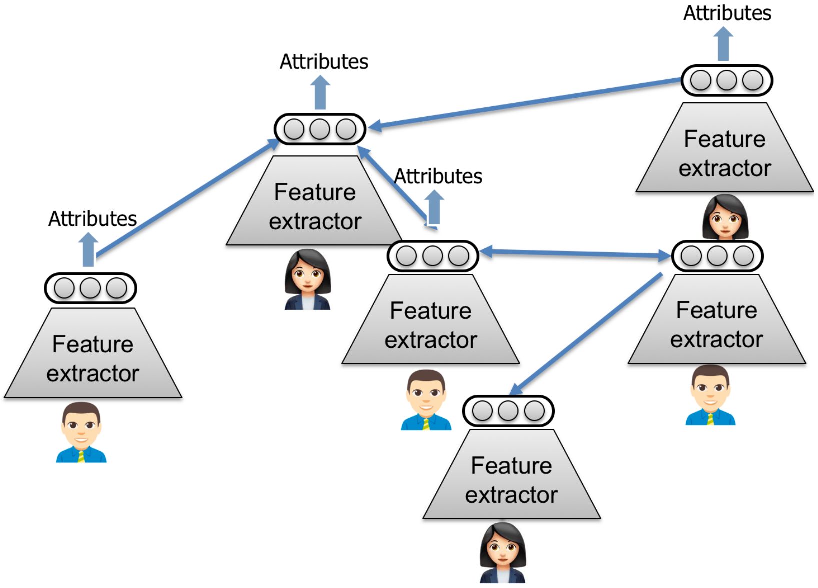


Fig. 5.1 An illustration of the graph-level attributes learning process.

An illustration of the learning process is shown in Fig. 5.1, where users' local features are first computed by a feature extractor function, then these features are propagated on the graph via a graph neural network.

In addition, such feature propagation procedure can not capture global characteristics of nodes in a social media. For example, users with millions of followers may not share much common properties with their followers. Incorporating classic network metrics like degree centrality, betweenness centrality may be a possible way to alleviate this issue.

5.2 Related Work

The concept of graph neural networks was first introduced in [101]. Given a graph with adjacent matrix $A \in R^{N \times N}$, the representation h_i for a node i is updated as follows:

$$h_i = f(x_i, x_{e[i]}, h_{n[i]}, x_{n_i}) \tag{5.4}$$

$$o_i = g(h_i, x_i) \tag{5.5}$$

where x_i, $x_{e[i]}$, $h_{n[i]}$, x_{n_i} are features of node i, features of its edges, the states, and the features of its neighbourhood. Function f is a contraction map and are shared across layers. The final representation h_i for node i is a fixed point of f. Combining h_i and x_i, it outputs label o_i for node i. In general, this process can be viewed as features propagation from neighbourhood.

There are several GNN variants exist in the literature. Kipf and Welling introduce a simplified spectral approach called graph convolutional neural networks (GCN) [65]. They use one-step neighbourhood to update the state of a central node as:

$$H^{l+1} = \hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H^l \Theta^l \tag{5.6}$$

where $\hat{A} = A + I$, $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$, $H^l \in R^{N \times C_l}$ is the stacked states for all nodes at layer l, H^0 is stacked node features X, $\Theta^l \in R^{C_l \times C_{l+1}}$ is a filter parameter. C_l is the dimension of hidden states at layer l.

Another popular variant is the graph attention network (GAT) [113]. Again, a node's state is updated by aggregating its neighbourhood's states. GAT adopted one widely used multi-head attention method in natural language processing to learn important nodes in neighbourhood [112]. Using K attention heads, GAT update states by

$$h_i^{l+1} = \mathop{\Big|\Big|}_{k=1}^{K} \sigma\Big(\sum_{j \in n[i]} \alpha_{ij}^{lk} W^{lk} h_j^l\Big) \tag{5.7}$$

$$\alpha_{ij}^{lk} = \frac{exp(LeakyReLU({a_k^l}^T [W^{lk} h_i^l || W^{lk} h_j^l]))}{\sum_{u \in n[i]} exp(LeakyReLU({a_k^l}^T [W^{lk} h_i^l || W^{lk} h_u^l]))} \tag{5.8}$$

where $||$ represents vector concatenation, α_{ij}^{lk} is the attention coefficient of node i to its neighbour j in attention head k at layer l. $W^{lk} \in R^{\frac{C_{l+1}}{K} \times C_l}$ is a linear transformation for input states. σ denotes a sigmoid function. $a_k^l \in R^{\frac{2C_{l+1}}{K}}$ is an attention context vector learned during training.

In practise, researchers have observed that deeper GNN models could not improve performance and even perform worse, which is partially due to more layers would also propagate noisy information from expanded neighborhood [65]. A common option is using a residual connection as shown in Eq. 5.9, which adds states from lower layer directly to higher layer and avoids the local features getting vanished in higher layers.

$$H^{l+1} = GNN(H^l, A; \Theta^l) + H^l \tag{5.9}$$

where Θ_l is parameter of GNN at layer l.

5.3 Method

We treat each user as a unique node in a social network, where users are connected via certain social relationships, eg. following, mentioning. Each user is associated with local raw text features such as tweet texts and profile description. Based on the network connections and users' local text features, the ultimate task is to infer users' certain latent attributes such as their political leaning and social identities.

Formally, given there are n users in the social network, we denote the graph as an adjacent matrix $A \in R^{n\times n}$. For user i, there are m text features in total, which can be written as $T_{i1}, T_{i2}, ..., T_{im}$. Each text field is a sequence of words such as $T_{ij} = [w_1, w_2, ..., w_l]$. The goal is to classify each user into an attribute category.

Figure 5.1 shows the overall architecture of our framework. For users in the social network, they share a common task-specific feature extractor and this feature extractor learns high-level feature representations from these raw text features. Based on these learned high-level local representations, we employ a graph neural network (GNN) to propagate and aggregate these representations. The final hidden representations generated by the GNN are used for the final attribute prediction. We will elaborate each component of our method in the following subsections.

5.3.1 Feature extractor

For words in each text field, we first map them into a word embedding space R^D. A text field $T_{ij} = [w_1, w_2, ..., w_l]$ can be transformed into a sequence of embedding vector $[x_1, x_2, ..., x_l]$, where x_l is the word embedding vector for word w_l.

We then apply a Bi-LSTM neural network to extract high-level semantic representations from text embedding matrices. At every time step t, a forward LSTM takes the word embedding x_t of word w_t and previous state $\overrightarrow{h_{t-1}}$ as inputs, and generates the current hidden state $\overrightarrow{h}_t$. A backward LSTM reads the text from w_l to w_1 and generates another state sequence. The hidden state $h_t \in R^{2D}$ for word w_t is the concatenation of $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$. Concatenating all the hidden states, we get a semantic matrix $H \in R^{l\times 2D}$

$$\begin{aligned}
\overrightarrow{h_t} &= \overrightarrow{LSTM}(x_t, \overrightarrow{h_{t-1}}) \\
\overleftarrow{h_t} &= \overleftarrow{LSTM}(x_t, \overleftarrow{h_{t+1}}) \\
h_t &= [\overrightarrow{h_t}, \overleftarrow{h_t}]
\end{aligned} \tag{5.10}$$

The text representation of text field is a weighted average of the word semantic representations. This is achieved by a multi-head attention layer [112]. In each attention layer, there are h attention heads which learn to attend important information. Each head computes a text representation as a weighted average of these word hidden states. The computation steps in a multi-head attention layer are as follows.

$$\begin{aligned}
f = MultiHead(q,H) &= [head_1,...,head_h]W^O \\
head_i(q,H) &= softmax(\frac{qW_i^Q \cdot (HW_i^K)^T}{\sqrt{d_k}})HW_i^V
\end{aligned}$$

where $q \in R^{2d}$ is an attention context vector learned during training, $W_i^Q, W_i^K, W_i^V \in R^{2D\times d_k}$, and $W^O \in R^{2D\times 2D}$ are projection parameters, $d_k = 2D/h$. An attention head $head_i$ first projects the attention context q and the semantic matrix H into query and key subspaces by W_i^Q, W_i^K respectively. The matrix product between query qW_i^Q and key HW_i^K after softmax normalization is an attention weight that indicates important words among the projected value vectors HW_i^V. Concatenating h heads together, we get one representation vector $f \in R^{2D}$ after projection by W^O for each text field.

Given m text fields, we get m text representation vectors $F_i = [f_{i1}, f_{i2}, ..., f_{im}]$ for user i. We further use several layers of transformer encoders [112] to learn the correlation between different feature fields. Each layer consists of a multi-head self-attention network and a feed-forward network (FFN). One transformer encoder layer first uses input feature to attend important information in the feature itself by a multi-head attention sub-layer. Then a linear transformation sub-layer FFN is applied to each position identically. Same as [112], we employ residual connection [49] and layer normalization [6] around each of the two sub-layers. The output F_1 of the first transformer encoder layer is generated as follows.

$$\begin{aligned}
F' &= LayerNorm(MultiHead(F,F) + F) \\
F_1 &= LayerNorm(FFN(F') + F')
\end{aligned}$$

where $FFN(F') = max(0, F'W_1 + b_1)W_2 + b2$, $W_1 \in R^{2D \times D_{ff}}$, and $W_2 \in R^{D_{ff} \times 2D}$.

Since there is no position information in the transformer encoder layer, our model cannot distinguish between different types of features, eg. tweet text and personal description. To overcome this issue, we add feature type embeddings to the input representations F. There are m text fields in total. Each of them has a learned feature type embedding with dimension $2D$ so that one feature type embedding and the representation of the corresponding feature can be summed.

Because the input and the output of transformer encoder have the same dimension, we stack L layers of transformer encoders to learn representations for the attribute prediction. The final classification features are the row-wise weighted average of F_L. Similar to the word-level attention, we use a field-level multi-head attention layer to select important features from m vectors and fuse them into a single vector.

$$v = MultiHead(q_f, F_L)$$

where $q_f \in R^{2D}$ is an attention context vector.

5.3.2 Graph-level feature aggregation

A graph attention network (GAT) [113] is a variant of graph neural network [101] and is a key element in our method. It propagates features from a user's friends to this user. Given a social graph with n nodes, each user's local feature is generated by the feature extractor as described in the previous section. One GAT layer compute node representations by aggregating neighbourhood's hidden states. With an L-layer GAT network, features from L hops away can be propagated to one user.

Specifically, given a user i with a hidden state v_i^l at layer l and the node's neighbours $n[i]$ as well as their hidden states, a GAT updates the node's hidden state at layer $l+1$ using multi-head attentions [112]. The update process is as follows

$$v_i^{l+1} = \Big\Vert_{k=1}^{K} \sigma\Big(\sum_{j \in n[i]} \alpha_{ij}^{lk} W_{lk} h_j^l\Big) \tag{5.11}$$

$$\alpha_{ij}^{lk} = \frac{exp(f(a_{lk}^T[W_{lk}h_l^i||W_{lk}h_j^l]))}{\sum_{u \in n[i]} exp(f(a_{lk}^T[W_{lk}h_i^l||W_{lk}h_u^l]))} \tag{5.12}$$

where $||$ represents vector concatenation, α_{ij}^{lk} is the attention coefficient of node i to its neighbour j in attention head k at layer l. $W_{lk} \in R^{\frac{D}{K} \times D}$ is a linear transformation matrix for input states. D is the dimension of hidden states. σ denotes a sigmoid function. $f(\cdot)$ is a LeakyReLU non-linear function [71]. $a_{lk} \in R^{\frac{2D}{K}}$ is an attention context vector learned during training.

For simplicity, we can write such feature propagation process as

$$V^{l+1} = GAT(V^l, A; \Theta_l) \tag{5.13}$$

where $V^l \in R^{N \times D}$ is the stacked states for all nodes at layer l, $A \in R^{N \times N}$ is the graph adjacent matrix. Θ_l is the parameter set of the GAT at layer l.

We further utilize an LSTM unit to model long-term dependency across layers, which is also helpful for overcoming noisy information in a graph [57]. At each layer $GAT(V^l, A; \Theta^l)$ generates new input for an LSTM unit, and this LSTM unit decides how much information should be added into the next layer.

$$\begin{aligned} V^0, C^0 &= LSTM(V, (0, 0)) \\ V^{l+1}, C^{l+1} &= LSTM(GAT(V^l, A; \Theta_l), (V^l, C^l)) \end{aligned}$$

Note that such feature propagation procedure does not take network structure information into consideration explicitly. This feature aggregation procedure is the same for users with million of followers and users with hundreds of followers. However, a users with millions of followers may not share much common properties with their followers. To further incorporate network structure information explicitly, we combine classic network metrics as additional input. These selected network metrics are as follow:

In-degree centrality: the number of in-coming edges for each node.

Out-degree centrality: the number of out-coming edges for each node.

Eigenvector centrality: the eigenvector centrality for node i is the i-th element of the eigenvector x with the largest eigenvalue λ. This centrality is measured by looking at nodes' followers. $Ax = \lambda x$.

K-core score: The k-core is found by recursively pruning nodes with degrees less than k. **Clustering coefficient**: The proportion of links between the nodes within its neighbourhood divided by all the possible links between them. **Reciprocity**: The ratio of links in both directions to the total number of edges.

For each graph metric, we first normalize them to zero mean and unit variance, then concatenate them with the original GAT input vector V.

5.3.3 Final Classification

With L layers of GAT networks, we get final representations for our target users. We just retrieve the corresponding hidden state v_i^L for the node.

We map the hidden state v_i^L into the classification space by a linear transformation. Afterwards, the probability of a attribute class c is computed by a softmax function:

$$P(y=c) = \frac{exp(Wv_i^L + b)_c}{\sum_{k \in C} exp(Wh_i^L + b)_k} \tag{5.14}$$

where W, b are the weight matrix and bias for the linear transformation, C is the set of attribute classes.

The final predicted attribute of a user is the label with the highest probability. We minimize the cross-entropy loss to train our model

$$loss = -\sum_{c \in C} I(y=c) \cdot log(P(y=c))$$

where $I(\cdot)$ is an indicator function.

5.4 Experiments Setting

5.4.1 Implementation details

For our base feature extractor, we choose 300-dimensional GloVe vectors [83] to initialize our word embeddings. These embeddings are adapted during training. The batch size is set as 32. We train our model with initial learning rate 10^{-4} with Adam update rule [64]. If the validation accuracy is not increased in one epoch, we decrease the learning rate to 10^{-5}. We set the number of heads in multi-head attention layers as 5. We apply dropout on the output of multi-head attentions and feed-forward layers in transformer encoders with dropout rate 0.1.

We use same number of attention heads in the GAT layers. We stack two layers of GAT in our graph-level feature aggregation. Practically, training for a large-scale social graph with millions of users becomes unfeasible because of the memory limitation. We use the sampling method proposed in GraphSAGE [44] for batched training. At each training iteration, we first sample a small batch of nodes B_0 and then recursively expand B_l to B_{l+1} by sampling S_l neighbourhood nodes of B_l. With a GNN of M layers, we get a hierarchy of nodes: $B_0, B_1, ..., B_M$. Representations of target users B_0 are updated by aggregating node states from the bottom layer B_M to the upper layer B_0. In addition, training the feature extractor and feature aggregator end-to-end requires a lot of GPU memory. We further propose to first train a local feature extractor, then update the parameters in the graph-level feature aggregator. Using such two-steps updating function saves a lot of computation cost while maintains a similar performance empirically.

5.4.2 Datasets

We adopt two datasets released in previous work to evaluate our method. The first is a public figure identification dataset for Twitter user [58], which is automatically constructed based on Twitter's verify field. The goal is to detect public figures based on their posts. The second is an identity classification dataset[58]. Annotators are asked to label Twitter users into seven identity classes. We also compiled one large scale political leaning dataset by using users' overt following ties. We select political figures unambiguously from both liberal and conservative politics. These Twitter users who only follow liberal politics are labeled as liberal, and who only follow conservative politics are labeled as conservative. We denote this dataset as Politic. We collected their most recent 200 tweets and following ties during October 2019. We remove all these political figures' accounts in the collected dataset. Otherwise, the task will become trivial. Statistics of these two datasets are shown in Table 5.1. For the relative small datasets Identity, we run our method in a semi-supervised setting, where users in this dataset are connected to users in the Public figure dataset.

In addition, we collected 2,052,708 unlabeled users to further enhance our method in the semi-supervised setting. We first used Twitter's streaming API to search tweets mentioning "coronavirus" between Jan. 29 and Jan. 31, then collected the timelines and followees of users who posted these tweets.

Table 5.1 Statistics of three datasets

Dataset	# of users			# of classes	# of edges
	Train	Dev.	Test		
Public figure	518K	5K	10K	2	69,567,290
Identity	11846	500	1000	7	69,567,290
Politic	791K	99K	99K	2	11,797,711
Unlabeled		2M			611,952,012

5.5 Results

As shown in Table 5.2, we compare our graph-level user attribute prediction system to a previous user-level prediction system [58]. Our system outperforms previous method in a large margin across all three datasets. We also compare our graph-level learning framework to Deepwalk [84], which is a network embedding method. As shown in the table, using network feature alone works worse than user-level method. Combining user-level model with Deepwalk indeed improves its performance. However, the performance is still worse than our graph-level learning framework, especially on the identity dataset and politic dataset. In addition, we also enhance our method by adding two million unlabeled users, which is denoted as graph-level+. In this semi-supervised setting, we gain further improvement using unlabeled data.

Table 5.2 Results of our method on three datasets compared to previous method. Methods ending with "+" symbol are enhanced by 2M unlabeled users.

Method	Public figure		Identity		Politic	
	Acc.	F1	Acc.	F1	Acc.	F1
User-level	94.2	93.1	91.6	88.6	78.4	78.4
Deepwalk	80.5	72.4	78.9,	66.5	78.2	78.3
User-level+Deepwalk	96.3	95.5	92.2	89.6	81.5	81.4
Graph-level	**97.8**	**97.4**	**94.0**	**91.7**	**83.6**	**83.6**
Deepwalk+	82.6	76.1	79.1	70.7	84.2	84.1
User-level+Deepwalk+	96.1	95.3	92.2	89.5	84.7	84.6
Graph-level+	**98.1**	**97.7**	**95.4**	**93.5**	**87.4**	**87.4**

5.5.1 Importance of network metrics

We plot the performance loss on the identity classification task when we remove each network centrality one by one in Figure 5.2. As shown in this figure, the classification accuracy and F1-score drop by 1.6% and 2.7% respectively after removing all the network metrics. In-degree centrality is the most important feature followed by reciprocity and k-core score. These network metrics provide additional network structure information to the graph neural networks.

5.5.2 Sampling size

When we train our graph-level prediction system end-to-end, the entire feedforward and backpropagation finish in one single step. The gradient would be back-propagated

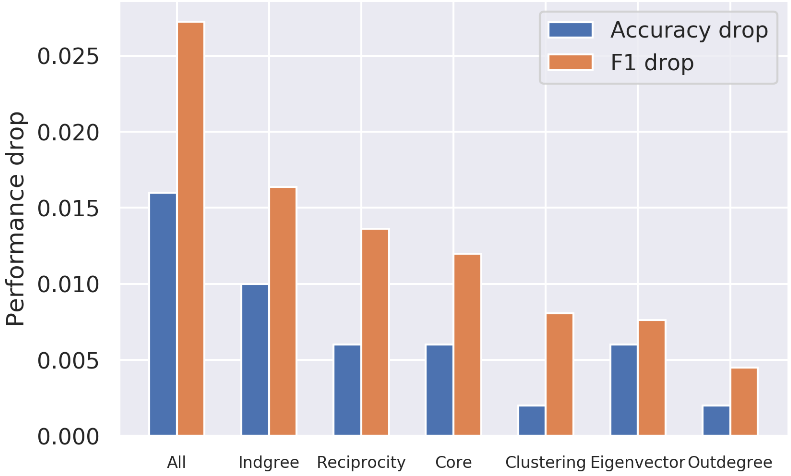


Fig. 5.2 The performance loss when we remove each network centrality.

from the final classification layer to the graph neural network layer and then to the user-level feature extractor layer, which requires a lot of GPU memory. We use 4 Titan Xp GPUs for the end-to-end training experiment. Practically, we can only sample 10 neighbourhood for each user with a batch size of 32, while training time is increased by a factor of 10. Instead, we first train a local user-level classifier, then cache the user-level feature representations and use them as input for the graph-level aggregator in the second-phase training. In Figure 5.3, we show the model performance with respect to the sampling size. The dashed green line represents the performance when we train our method end-to-end with 10 sampled neighbours. The blue line represents the performance of our two-step training with sampling size B_1 ranging from 5 to 50. The orange line shows the performance of two-step training with two-layer sampling. In the first hop, we sample B_1 direct connected neighbours with B_1 ranging from 5 to 50. In the second hop, we sample 10 neighbours for each users in B_1. As shown in this figure, with the same sampling size, the two-step training generally works worse than end-to-end training. However, we can easily scale up our sampling size in the two-step training and gain additional performance increase.

5.5.3 Scalability

Scalability is always a concern when we apply our machine learning model on a massive of users in a large scale social network. Assume each user has in total of T tweets, the time complexity to compute the user feature is $O(T)$. For a two-layer sampling with sampling sizes B_1 and B_2, it takes $O(T + T \times B_1 \times B_2)$ to compute the user-

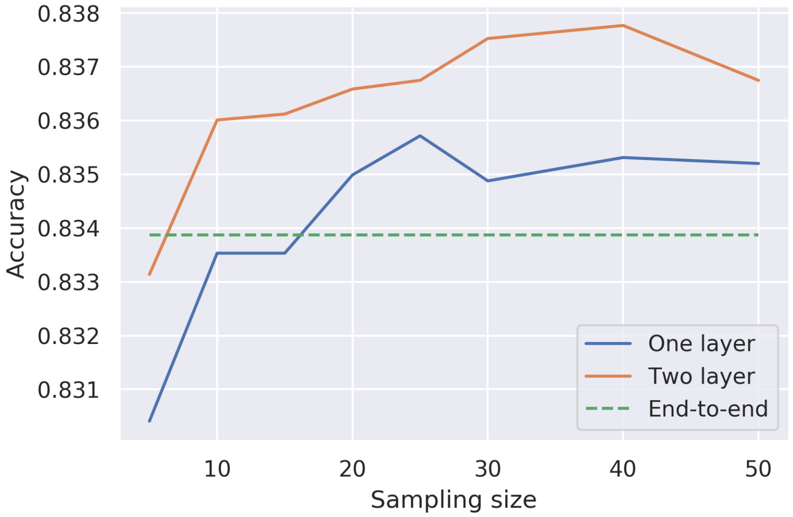


Fig. 5.3 Model performance with respect to the sampling size on the Politic dataset.

level features for all the sampled users. In practise, this user-level feed-forward pass takes much longer time than the graph-level aggregation step. Hence, when we apply our method on large scale dataset, we cached all the user-level representations in a database. As a result, given a lot of users' representations cached in the database, the time complexity of the user-level feed-forward pass for a new incoming user can be reduced from $O(T + T \times B_1 \times B_2)$ to $O(T)$.

In the Figure 5.4, we show the tradeoff between the model performance and runtime consumption. As shown in the figure, using only one tweet would greatly reduce the inference time by more than 5 times compared to using user's whole timeline. In the meanwhile, though with a longer computation time, the user-level model improves the performance in a large margin. Last, we can see aggregating user's friends' features also improve the performance with the cost of additional computation overhead.

5.6 Conclusion

In this paper, we present a user attribute prediction system which incorporates users' local textual features as well as the social graph. We first extract user representations from local text fields by a deep neural network, then aggregate neighbours' features to generate final predictions. We achieve the state-of-the-art results on three benchmark datasets. We further demonstrate that under semi-supervised setting we can greatly improve our system's performance by adding unlabeled users.

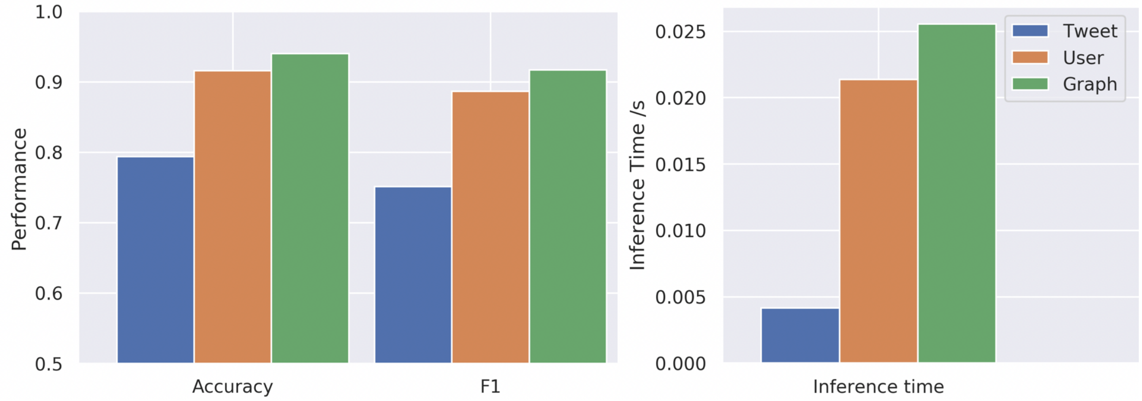


Fig. 5.4 Tradeoff between model performance and runtime on the identity dataset.

Certain limitations also exist in this paper. First, we assume users' attributes are static overtime. However, certain attributes such as political orientation, social roles may evolve. Future work should take the attributes dynamic into consideration. In addition, users may hold different political ideology under different circumstances and political topics. In the future, we plan to tackle this issue by looking at aspect-specific political orientation prediction. Hence, we can get users' stances of different issues. Given a pair of input — user's profile and a political issue, we can output a stance of each user for this specific issue.

Besides, this work only utilizes the following network among Twitter users. In reality, there are many other social interactions among users, eg. mentioning, retweeting. In our experiments, we also tried propagate features both from following friends and mentioned users separately. However, the mentioning graph is much sparser than the following graph. In the case of public figure dataset, the density of following graph is 0.25‰, which is 6.6 times as high as as the density of mentioning graph. As a result, the additional information from mentioned friends does not provide much improvement. Features propagated from following friends dominate in the classification process. In the future, we would like to tackle this problem and find out a way to handle this multi-model graph issue.

Chapter 6

An Empirical Study of the Novel Coronavirus Outbreak on Twitter

6.1 Introduction

As 2020 began, an outbreak of a new respiratory disease that would come to be known as COVID-19 occurred. The disease was first reported from Wuhan, China on December 31, 2019 [1]. On January 30, 2020 the World Health Organization (WHO) declared the outbreak a public health emergency of international concern. By May 7, 2020, more than 3,900,000 cases were confirmed worldwide spread across 214 countries and regions, and 270,057 people had died. Severe outbreaks has occurred in China, United States, and Europe.

As the novel coronavirus spread globally, a growing public panic was expressed over the internet. We tracked this panic using Twitter. Twitter is one of major social media platforms where users expressed concerns about the outbreak of this disease, shared purported preventions and cures, discussed theories about where the disease came from, and how governments were and should respond. A significant fraction of the information being shared was "fake" as noted by numerous news agencies reports[2]. Online fact checking sites, like Poynter[3], put up new information each day about new disinformation stories. Our analysis of these stories[4] showed many types of disinformation stories: false preventions and cures, false claims about the nature of the

[1] https://www.who.int/emergencies/diseases/novel-coronavirus-2019

[2] https://www.nytimes.com/2020/03/08/technology/coronavirus-misinformation-social-media.html?searchResultPosition=1

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

[4] https://www.cmu.edu/ideas-social-cybersecurity/research/index.html

disease, false diagnostic procedures, false origin stories, false emergency measures, false "feel good" stories, and so on.

The research community has frequently turned to social media to study the spread of information, disinformation and misinformation [3, 68, 69, 8]. Among the platforms studied are: Twitter [114, 102, 7], Facebook [80, 14, 32], and Youtube [35, 87, 70]. Disinformation or "fake news" has recently draw attention primarily in a political context, such as studies around elections [41, 3, 111]. Bovet and Makse investigated 30 million tweets containing a link to news outlets preceding the election day. They found that the top influencers spreading traditional center and left leaning news largely influence the Clinton supporters, while top "fake news" spreaders influence Trump supporters [17]. Grinberg et al. found that only small portion of individuals accounted for a majority of "fake news" sharing [41]. They also found conservative leaning users are more likely to engage with "fake news" sources. "fake news" also emerges in information about topics such as vaccination and natural disaster. Chiou and Tucker studied the role of social media in the dissemination of false news stories about vaccines. They documented that members of anti-vaccine Facebook groups tended to disseminate false stories beyond the group through diverse media. Gupta et al. in a study of fake images during Hurricane Sandy[42] found that the top thirty users resulted in 90% of retweets of fake images. Most prior research has focused on specific users, with little concern for the type of user or their geographic location. An exception here is the work by Babcock and colleagues that shows that disinformation spread by celebrities or newsagencies has greater reach [8], and Carley et al. [23] that news agencies are typically the most retweeted users, particularly during disasters.

During a pandemic, trust in health authorities is critical to prevent the spread of the disease, to save lives, and to enable public safety. Misinformation is damaging and can even be deadly. Because of the severe consequence, it is critical to understand the spread of accurate and inaccurate information. A key problem in a global pandemic is that while these authorities, other than the World Health Organization are local, information and disinformation is spread globally. Hence, disinformation from one country can undermine, even unintentionally, the heath authority in another country. This may be particularly true when the information appears to come from a credible source such as a newsagency or government official. However, little is known about the spread of information, let alone misinformation, between countries. Little is known about the role of types of actors, such as newsagencies, in the spread of information and disinformation particularly from a global perspective. In this study, we examine the global spread of information related to key disinformation stories during the early stages of the global pandemic. We address four research questions:

1. What types of users send influential tweets in this global health emergency event?
2. Who is discussing disinformation stories?
3. Where in the world are those who discuss low credibility information?
4. What is the global network for discussing low credibility information?

6.2 Data and definitions

6.2.1 Data collection

To answer these questions, we monitored conversations about COVID-19 on Twitter starting from January 29, 2020 to March 4, 2020. We select a list of keywords to track Twitter's real-time conversations[5]. The list include "coronavirus", "coronaravirus", "wuhan virus", "wuhanvirus", "2019nCoV", "NCoV", "NCoV2019". There are 67.4 million tweets and 12.0 million users involved in this time period. We recognize that most Twitter users are tweeting in English, and that much of the discussion around the coronavirus early on, used the English terms if it was on Twitter. Hence, we used these predominantly English keywords, and the WHO terms. Although this creates a bias toward the spread of English tweets, it does pick up a large number of tweets in other languages.

In Figure 6.1, two red lines show the general trends for number of tweets and number of users involved each day. The two blue lines represent the number of newly confirmed patients each day in China as well as those outside of China reported by WHO [6]. As shown in this figure, the volume of tweets first gradually reduced as the disease was contained inside China. As confirmed cases spread throughout the world, the volume of daily conversation in Twitter soared. By February 26, it was six times higher than on February 20 (3,904,293 v.s. 638,204).

To further determine users' location, social identity, and political orientation, we also collected the most recent 200 tweets and the following ties for each user who posted tweets between January 29, 2020 and March 4, 2020. Among 12,047,990 involved users, we successfully collected information for 11,951,739 users. These data are further fed into a state-of-the-art user profiling system [55, 58]. Using this system we predict the users' home country based on these tweets with 92.96% accuracy. We classify users' social identity into seven categories – news media, news reporter, celebrity, government

[5] https://developer.twitter.com/en/docs/tweets/filter-realtime/guides/basic-stream-parameters

[6] https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports

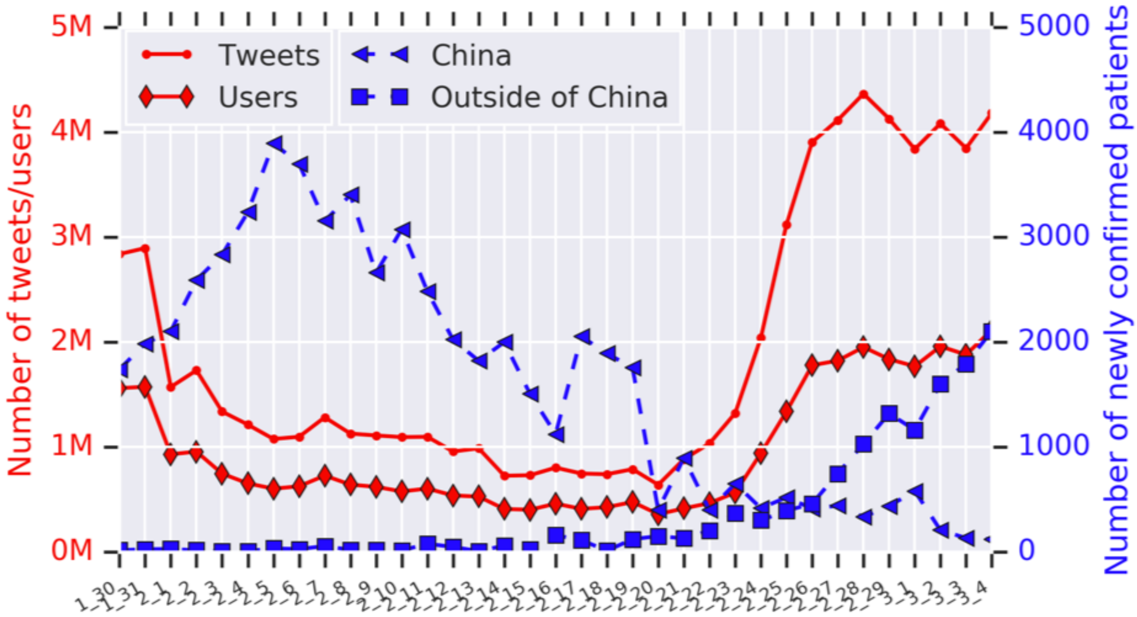


Fig. 6.1 The red lines show the number of tweets and number of users each day. The blue lines represent the newly confirmed patients each day.

official, sport, company, and regular user. We also predict users' political orientation as liberal or conservative. We achieve accuracies of 95.4% and 87.4% on these standalone test datasets for identity classification and political orientation prediction respectively. Because our identity and political orientation classifiers are mainly trained on English users, we only apply these two classifiers to users whose major language is English.

6.2.2 Disinformation and misinformation

Following previous work [68] and [41], we define "fake news sites" as ones that "lack the news media's editorial norms and processes for ensuring the accuracy and credibility of information." We adopt three lists of fake news sites as proposed in [41]. The black list contains a set of websites which published exclusively fabricated stories. The red list is a set of websites spreading falsehoods with a flawed editorial process. Sites labeled as orange represent cases where annotators were less certain that the falsehoods stemmed from a flawed editorial process. We further add a list of news sources as trusted news sites. There are 20 black, 26 red, 25 orange fake news sites and 90 real news sites whose URLs appear in our collected data.

To study the conversation around specific disinformation stories, we manually identified five disinformation story-lines. The first is a popular conspiracy that this

Source	Total	Bio-weapon	Bleach	Chlorine	Garlic	Sesame
Black	3083	92 (2.98%)	0 (0.00%)	0 (0.00%)	0 (0.00%)	0 (0.00%)
Red	53522	4,447 (8.31%)	0 (0.00%)	0 (0.00%)	5 (0.01%)	0 (0.00%)
Orange	61162	4,879 (7.98%)	0 (0.00%)	1 (0.00%)	10 (0.02%)	0 (0.00%)
Real	796071	1,245 (0.16%)	0 (0.00%)	13 (0.00%)	169 (0.02%)	18 (0.00%)

Table 6.1 Number of source tweets with news URLs, overall and by storyline.

Retweet	Total	Bio-weapon	Bleach	Chlorine	Garlic	Sesame
Black	32302	73 (0.23%)	0 (0.00%)	0 (0.00%)	0 (0.00%)	0 (0.00%)
Red	151249	15,467 (10.23%)	0 (0.00%)	0 (0.00%)	2 (0.00%)	0 (0.00%)
Orange	205362	24,916 (12.13%)	0 (0.00%)	0 (0.00%)	11 (0.01%)	0 (0.00%)
Real	2738551	4,291 (0.16%)	0 (0.00%)	0 (0.00%)	205 (0.01%)	25 (0.00%)

Table 6.2 Number of retweets with news URLs, overall and by storyline.

novel coronavirus is a bio-weapon developed in a research lab. The remaining story-lines are about potential cures for this disease – garlic, sesame oil, bleach, and chlorine dioxide. For each story-line, we retrieve tweets by searching for corresponding keywords in our COVID-19 corpus. For this body of tweets, we can say that they are discussing a particular disinformation story-line; however, at this point we cannot say whether or not the sender of the message is knowingly spreading disinformation, unwittingly spreading misinformation, joking about the story-line (satire), or pointing out that this story-line is not true and so countering disinformation. What we can say is that they are taking part in the discussion around that story-line.

We show the number of tweets that contain one of the news URLs and a story-line in Table 6.1 for original tweets and in Table 6.2 for retweets. As shown in Table 6.1, most of source tweets with fake news URLs contain keywords related to the bio-weapon conspiracy. There is a high percentage of tweets both mentioning “bio-weapon” and fake news URLs compared to tweets mentioning “bio-weapon” and a trusted news source. This suggests that the bio-weapon disinformation came from these fake news sites. In the case of retweets, a high percentage of retweets that mention red news or orange news sites also mention “bio-weapon”. This suggests that these less credible news sites were critical in further spreading this conspiracy; particularly as tweets mentioning black news URLs and “bio-weapon” were less likely to be retweeted. It is likely that there are more tweets discussing these stories, than listed here, as selection on keyword tends to under-sample.

6.3 Results

6.3.1 What types of users spread influential tweets?

There are 67,408,573 tweets posted by 12,047,990 users in total between January 29, 2020 and March 4, 2020. Among 5,448,250 English speaking users, 28.6% exhibit bot-like characteristics and behavior. There are 96.44% regular users (30.73% are labeled as bots[7]), 1.12% news agencies (32.91% of which are bots), 1.18% news reporters (18.04% of which are bots), and 1.05% government officials (19.94% of which are bots). Only 0.15% of users are company accounts and the rest appear to be celebrities. As companies and celebrities together make up less than 1% of the users, we do not continue to analyze these groups in this paper.

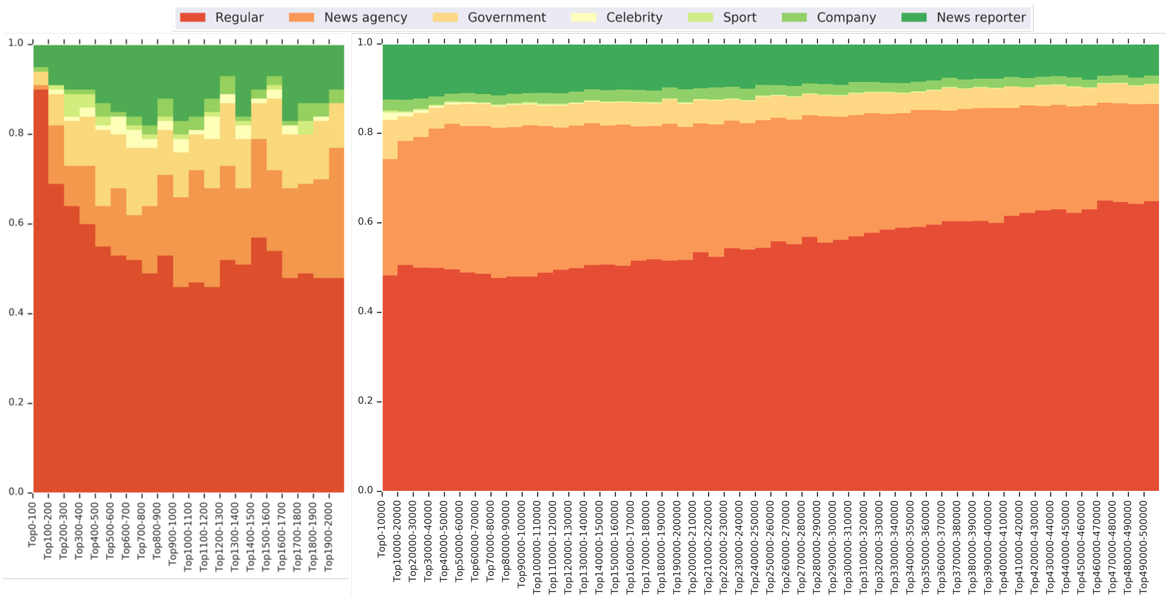


Fig. 6.2 Two bar charts show identity distributions across top influential tweets. There are 20 bars in the left figure, each of which represents the identity distribution of source users of 100 tweeters. The leftmost bar is for the top 100 most retweeted tweets. From left to right, the number of retweets decrease. The right figure contains 50 bars. Each bar shows the identity distribution of 10000 tweeters. The leftmost bar is the identity distribution for the most retweeted 10000 tweets.

We define influential tweets as those which were the most retweeted. For these influential tweets, we identify the type of user (identity) mainly tweeting in English that posted it. In Figure 6.2, two bar charts show the identity distribution across these most widely spread tweets. As expected, news medias play an important role in this

[7] We use bothunter with a 60% cutoff which has a precision of .957 and a recall of .704. [13, 12, 11]

event. 25.98% and 12.38% of top 10000 most influential tweets are posted by news agencies and individual news reporters. Government officials also contribute 8.79% of these 10000 influential tweets. Even though less than 5% of users are these official accounts, they contribute more than 50% of these top 10000 tweets. Regular users posted 48.29% of these influential tweets. The percentage is relatively small considering more than 95% of the users are regular users. A deeper dive into these top 100 most influential tweets, shows that 90% of the tweets are posted by regular users. Thus there is a curvilinear relationship with influence such that low influence and super high influence tweets are posted by regular users; whereas highly influential tweets are posted by news agencies and government officials.

6.3.2 What types of users cite “fake news” sites or discuss disinformation story-lines?

To study the identity of users who are talking about “fake news” URLs, we first retrieve all the tweets that contained a URL to a news site (fake or real), then we apply our user profiling system to these users. In total, there are 3085 source tweets containing black news URLs, 53531 source tweets containing red news URLs, and 61179 source tweets containing orange news URLs, and 796267 containing real news URLs. The number of retweets containing such URLs are 32305 (black), 151286 (red), 205409 (orange), and 2739257 (real).

In Table 6.3, we listed the number of tweets by each type of user that contain a URL to a news site by level of credibility. As shown in this table, news sites with different levels of credibility attract different types of users ($\chi^2 = 8832.1, p < 0.001, df = 18$). About 90% of the tweets containing these “fake news” URLs are initiated by regular users. Most of those tweets contain links to red or orange sites. In contrast, the real news sites are linked to by governments and individual news reporters.

We show the number and percentage of tweets sent by each type of user with bot-like behavior by levels of credibility in Table 6.4. We find that the lower the credibility of the news site being linked to, the more likely the sender of the tweet is a bot. For example, 58.74% of the source users sending a URL for a black news site appear to be bots as shown in Figure 6.3. A majority of retweeters of black sites are predicted as bot accounts, which indicates that a large group of automatically operated accounts are trying to promote these news. We also note that some of the news agencies, government and news reporter users appear to be bots. None of the accounts labeled as possible bots are verified accounts. There are several possible reasons for this: a) there are news bots and propaganda bots that are employed by some news agencies and various official account, b) an account where multiple people send out the tweets can appear as a bot, and c) despite 95.7% precision the bot-hunter program may be making errors. Nonetheless, the results suggest that there may be a set of bots that were established precisely to spread information from the less credible sites - particularly the black news sites.

	Regular	News agency	Government	News reporter
Black news	2,090 (81.45%)	409 (15.94%)	36 (1.40%)	27 (1.05%)
Red news	45,200 (91.64%)	3,155 (6.40%)	490 (0.99%)	357 (0.72%)
Orange news	45,503 (88.62%)	4,260 (8.30%)	821 (1.60%)	634 (1.23%)
Real news	509,188 (78.95%)	68,044 (10.55%)	15,430 (2.39%)	45,547 (7.06%)

Table 6.3 Number of tweets by each type of user that contain a URL to a news site by levels of credibility.

	Regular	News agency	Government	News reporter
Black news	1,487 (71.15%)	361 (88.26%)	29 (80.56%)	21 (77.78%)
Red news	30,165 (66.74%)	2,711 (85.93%)	294 (60.00%)	225 (63.03%)
Orange news	25,296 (55.59%)	2,392 (56.15%)	584 (71.13%)	295 (46.53%)
Real news	242,487 (47.62%)	41,633 (61.19%)	5,620 (36.42%)	10,912 (23.96%)

Table 6.4 Number of tweets by each type of user that also have bot-like behavior that contain a URL to a news site by levels of credibility, e.g. among 2090 tweets citing black news by regular users, 71.15% of them are posted by bot-like accounts.

What types of users are discussing the disinformation story-lines varies by story-line ($\chi^2 = 5233.8, p < 0.001, df = 30$) as can be seen in Table 6.5. The bio-weapon conspiracy is the most widely spread story-line as 34,301 unique users tweet about this story. Bleach is the most popular story-line concerning a false cure. A manual examination of the content of bleach tweets, showed that many of them appeared to be jokes or satirical responses to the original disinformation story. Fewer regular users spread the remaining story-lines. Many news agencies and government officials such as WHO are sending tweets trying to refute the original disinformation story-line.

In many cases, though, it appears that it is bots sending tweets regarding these disinformation story-lines. In Table 6.6 we see that there is not a simple pattern to the bot activity. We do find that there are more bot-like users spreading the bio-weapon story-line. This suggest that a set of bots may have been established to mimic authoritative sites to spread this information.

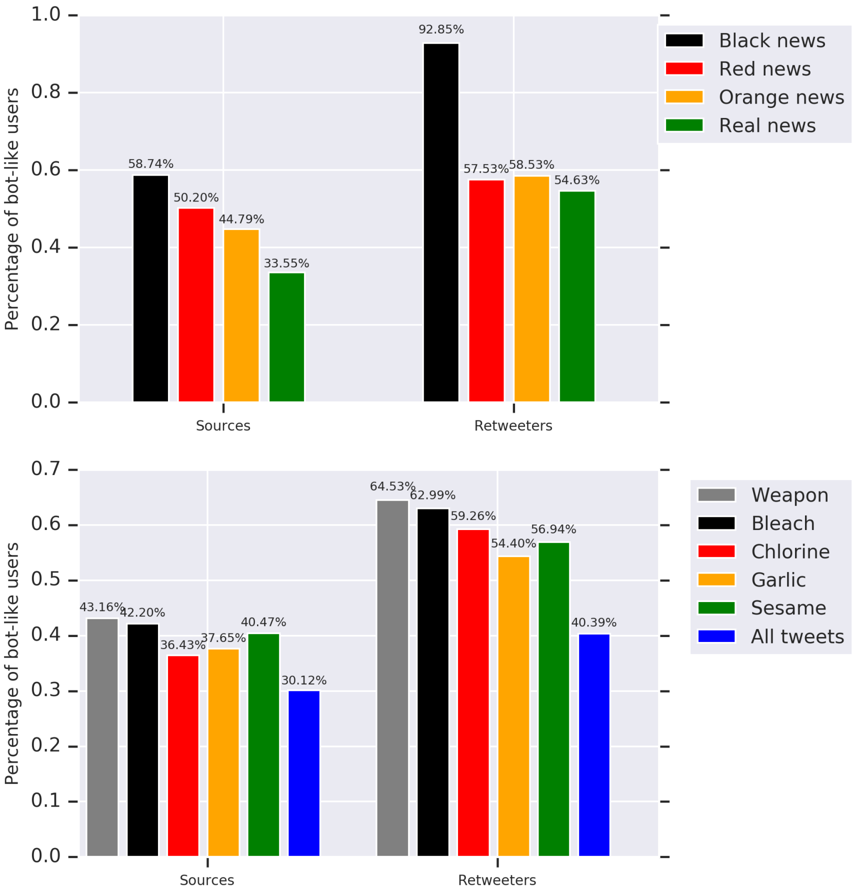


Fig. 6.3 Percentage of bot-like source users and retweeters who share news URLs (Above). Percentage of bot-like source users and retweeters who talked about misinformation (bottom). We also include same percentages of all the tweets in the chart below.

For the political orientation, Figure 6.4 shows most people who tweet and retweet “fake news” URL are more leaning towards conservative users. 33.28% of people tweeting real news URLs are labeled as conservative users, while 82.45% of people sharing “fake news” URLs are labeled conservative users. For each misinformation story, the percentage varies case by case. The widely spread bio-weapon story tends to attract a higher percentage of conservative users than liberal users. For the other cure stories, most of users discussing them are labeled as liberal users. Interestingly, even though chlorine dioxide is one kind of bleach, people who tweet about chlorine dioxide differ significantly from the bleach case, and the tweets are less likely to be jokes.

Misinformation	Regular	News agency	Government	News reporter
Bio-weapon	45,791 (91.54%)	1,956 (3.91%)	842 (1.68%)	1,192 (2.38%)
Bleach	5,826 (91.78%)	280 (4.41%)	72 (1.13%)	142 (2.24%)
Chlorine dioxide	262 (79.39%)	58 (17.58%)	4 (1.21%)	5 (1.52%)
Garlic	2,316 (79.89%)	385 (13.28%)	79 (2.73%)	100 (3.45%)
Sesame	279 (76.23%)	33 (9.02%)	27 (7.38%)	22 (6.01%)
All tweets	6,516,340 (79.96%)	1,102,842 (13.53%)	185,731 (2.28%)	263,939 (3.24%)

Table 6.5 Number of tweets by each type of user that mention one of the story-lines.

	Regular	News agency	Government	News reporter
Bio-weapon	21,556 (47.07%)	1,344 (68.71%)	590 (70.07%)	852 (71.47%)
Bleach	2,491 (42.76%)	134 (47.86%)	26 (36.11%)	31 (21.83%)
Chlorine	92 (35.11%)	13 (22.41%)	2 (50.00%)	3 (60.00%)
Garlic	946 (40.85%)	218 (56.62%)	23 (29.11%)	28 (28.00%)
Sesame	108 (38.71%)	22 (66.67%)	6 (22.22%)	4 (18.18%)
All tweets	2,759,427 (42.35%)	658,941 (59.75%)	72,451 (39.01%)	70,077 (26.55%)

Table 6.6 Number of tweets by each type of user with bot-like behavior that contain one of the story-lines, e.g. among 45,791 tweets mentioning bio-weapon by regular users, 47.07% of them are posted by bot-like accounts.

6.3.3 Where in the world are those who discuss low credibility information?

To measure users' geographical distribution, we combine geotags in user timelines with our country-level location predictions. For each user collected with timeline data, we look at the geotag in each tweet. Among 11,951,739 users, there are 791,830 individuals with geotags in their timelines. The home country of these geotagged users is determined by a majority vote of these geotags. For the remaining users, we apply our country-level location predictor to get their home countries.

Because these news lists are mainly compiled by English speakers, we only measured the country distributions among English speaking users to avoid potential language bias. As shown in Table 6.7, United States, United Kingdom, and Canada are the three countries from which most tweets with "fake news" URLs originate. Because of the language bias, there are many more tweets mentioning "fake news" and misinformation stories from countries mainly speaking in English such as United States and United Kingdom. 73.26% of the English speakers who posted tweets with "fake news" URLs came from United States. The percentage is also much higher than the U.S. users who posted real news URLs. To partially control for this bias, we normalize the number of users sharing "fake news" URLs by the total number of users in each country. The

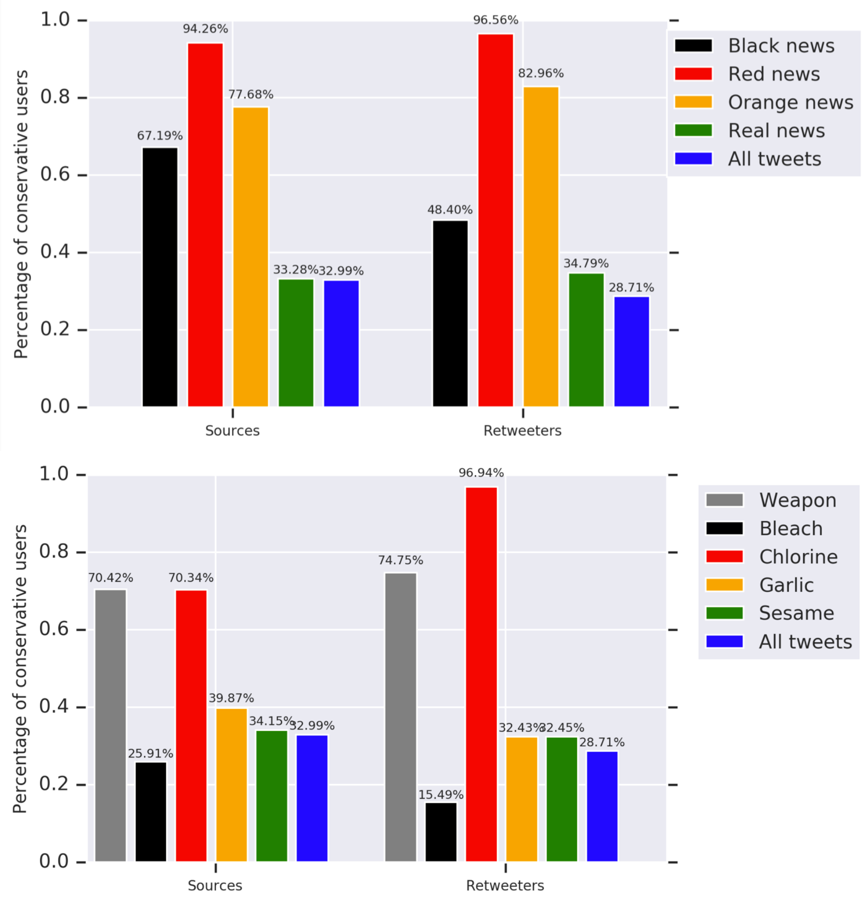


Fig. 6.4 Percentage of conservative source users and retweeters who share news URLs (Above). Percentage of conservative source users and retweeters who talked about misinformation (bottom). We also include same percentages of all the tweets in the chart below.

last column of Table 6.7 shows the normalized results. The probability for each U.S. user posting a "fake news" URL is 2.28%, while the probabilities for Philippines and Malaysia are 0.13% and 0.17%. Even though many users in United Kingdom shared "fake news" URLs, the normalized number is much lower than countries like the United States and Canada. Users mentioning "fake news" URLs are more different from the underlying population than users mentioning real news. We use KL-divergence to measure the distance from country distribution of source users posting "fake news" URLs to the country distribution of all English sources. The distance is 0.144 in this case. On the contrary, the KL-divergence from country distribution of sources posting real news URLs to the country distribution of English sources is only 0.075.

In Table 6.8, we also show the country distribution for users who retweet these "fake news" URLs. Again, a majority of retweeters are from the United States, followed

Country/region	# of EN users posting "fake news" URLs	# of EN users posting real news URLs	# of EN source users	% of EN users posting "fake news" URLs per country
United States	24,552 (73.26%)	146,875 (64.11%)	1,077,431 (51.55%)	2.28%
United Kingdom	2,599 (7.75%)	23,988 (10.47%)	255,779 (12.24%)	1.02%
Philippines	112 (0.33%)	2,483 (1.08%)	89,480 (4.28%)	0.13%
India	334 (1.00%)	4,674 (2.04%)	83,030 (3.97%)	0.40%
Canada	1,368 (4.08%)	10,228 (4.46%)	77,344 (3.70%)	1.77%
Nigeria	677 (2.02%)	1,263 (0.55%)	59,862 (2.86%)	1.13%
Australia	607 (1.81%)	6,755 (2.95%)	52,526 (2.51%)	1.16%
South Africa	339 (1.01%)	1,261 (0.55%)	26,308 (1.26%)	1.29%
Malaysia	39 (0.12%)	961 (0.42%)	22,869 (1.09%)	0.17%
Kenya	83 (0.25%)	900 (0.39%)	19,486 (0.93%)	0.43%
total	33,514	229,111	2,089,892	1.60%

Table 6.7 Country distribution of English speaking users who posted tweets with news URLs. We only show top 10 countries with the most English source users in this table.

Country/region	# of EN retweeters of "fake news" URLs	# of EN retweeters of real news URLs	# of EN retweeters	% of EN retweeters of fake news URLs per country
United States	79,157 (75.52%)	343,418 (64.18%)	2,088,187 (49.88%)	3.79%
United Kingdom	6,667 (6.36%)	49,222 (9.20%)	441,331 (10.54%)	1.51%
India	1,642 (1.57%)	13,277 (2.48%)	206,542 (4.93%)	0.79%
Philippines	241 (0.23%)	11,960 (2.24%)	204,856 (4.89%)	0.12%
Nigeria	2,953 (2.82%)	5,108 (0.95%)	135,306 (3.23%)	2.18%
Canada	3,586 (3.42%)	21,614 (4.04%)	135,083 (3.23%)	2.65%
Malaysia	380 (0.36%)	5,973 (1.12%)	118,361 (2.83%)	0.32%
Australia	1,393 (1.33%)	12,200 (2.28%)	69,699 (1.66%)	2.00%
Indonesia	98 (0.09%)	4,781 (0.89%)	62,147 (1.48%)	0.16%
South Africa	1,093 (1.04%)	2,875 (0.54%)	57,511 (1.37%)	1.90%
total	104,811	535,113	4,186,548	2.50%

Table 6.8 Country distribution of English speaking users who retweeted tweets with news URLs. We only show top 10 countries with the most English retweeters in this table.

by the United Kingdom and Canada. After we normalize the number of retweeters of "fake news" URLs by the total number of retweeters in each country, we can see that 3.79% of users in United States have at least retweeted one tweet with "fake news" URLs, which is much higher than the average percentage. Even though the United Kingdom has the second most English speaking retweeters, the probability for users in the United Kingdom retweeting "fake news" URLs is still lower than the average probability. Again, users retweeting "fake news" URLs are more different from the underlying population than are users mentioning real news. The KL-divergence from country distribution of retweeters of "fake news" URLs to English retweeters is 0.192, while the distance from real news retweeters to English retweeters is only 0.096.

We have similar observation on the country distribution of misinformation, as shown in Table 6.9. Most of users involved in the conversation about misinformation are from United States, followed by United Kingdom, Canada. Among these top 10 countries,

Country/Region	# of EN sources talking misinfo.	# of EN sources	% of EN sources talking misinfo. per country	# of EN retweeters of misinfo.	# of EN retweeters	% of retweeters of misinfo. per country
United States	23,234 (63.40%)	1,077,431 (51.55%)	2.16%	93,696 (67.54%)	2,088,187 (49.88%)	4.49%
United Kingdom	2,327 (6.35%)	255,779 (12.24%)	0.91%	7,484 (5.39%)	441,331 (10.54%)	1.70%
Philippines	1,142 (3.12%)	89,480 (4.28%)	1.28%	2,401 (1.73%)	204,856 (4.89%)	1.17%
India	1,351 (3.69%)	83,030 (3.97%)	1.63%	5,330 (3.84%)	206,542 (4.93%)	2.58%
Canada	1,541 (4.21%)	77,344 (3.70%)	1.99%	4,791 (3.45%)	135,083 (3.23%)	3.55%
Nigeria	1,080 (2.95%)	59,862 (2.86%)	1.80%	4,181 (3.01%)	135,306 (3.23%)	3.09%
Australia	879 (2.40%)	52,526 (2.51%)	1.67%	1,962 (1.41%)	69,699 (1.66%)	2.81%
South Africa	359 (0.98%)	26,308 (1.26%)	1.36%	1,539 (1.11%)	57,511 (1.37%)	2.68%
Malaysia	219 (0.60%)	22,869 (1.09%)	0.96%	2,009 (1.45%)	118,361 (2.83%)	1.70%
Kenya	341 (0.93%)	19,486 (0.93%)	1.75%	1,773 (1.28%)	39,558 (0.94%)	4.48%
total	36,645	2,089,892	1.75%	138,723	4,186,548	3.31%

Table 6.9 Country distribution of English speaking users who are involved in the conversation of misinformation. We only show top 10 countries with the most English source users in this table.

people from United States, Canada are more likely to tweet or retweet misinformation. Among 1,077,431 U.S. users who posted tweets, 2.16% of them posted tweets mentioning misinformation stories. And 4.49% of retweeters in U.S. have retweeted tweets talking misinformation stories. One thing we want to note is that even though only 11,734 and 32,643 English-speaking tweeters and retweeters are from Hong Kong, 2.86% tweeters and 6.99% retweeters from Hong Kong have tweeted or retweeted tweets mentioning misinformation phrases.

We plot the Kl-divergence between the country distribution of English speaking users who share specific news and the country distribution of all English speakers in Figure 6.5. As shown in this figure, the population of who sharing real news is the closest one to the underlying one. The less credible the news sites, the bigger of the population difference. In the bottom of Figure 6.5, we also show the same plot for all misinformation stories. The country distribution of source users who talked about the bio-weapon conspiracy are very close to the one of underlying population. Among the cure stories, the bleach misinformation is the one spread most closest to the underlying population.

6.3.4 What is the global network for discussing low credibility information?

An important question that has received little attention is how information about low credibility websites and disinformation story-lines spread between countries. To examine this, we extracted the information flow among countries. If a user in country A retweets a tweet posted by a user in country B, then we add an edge from country B to country A. Figure 5 shows the percentage of retweets between countries for tweets

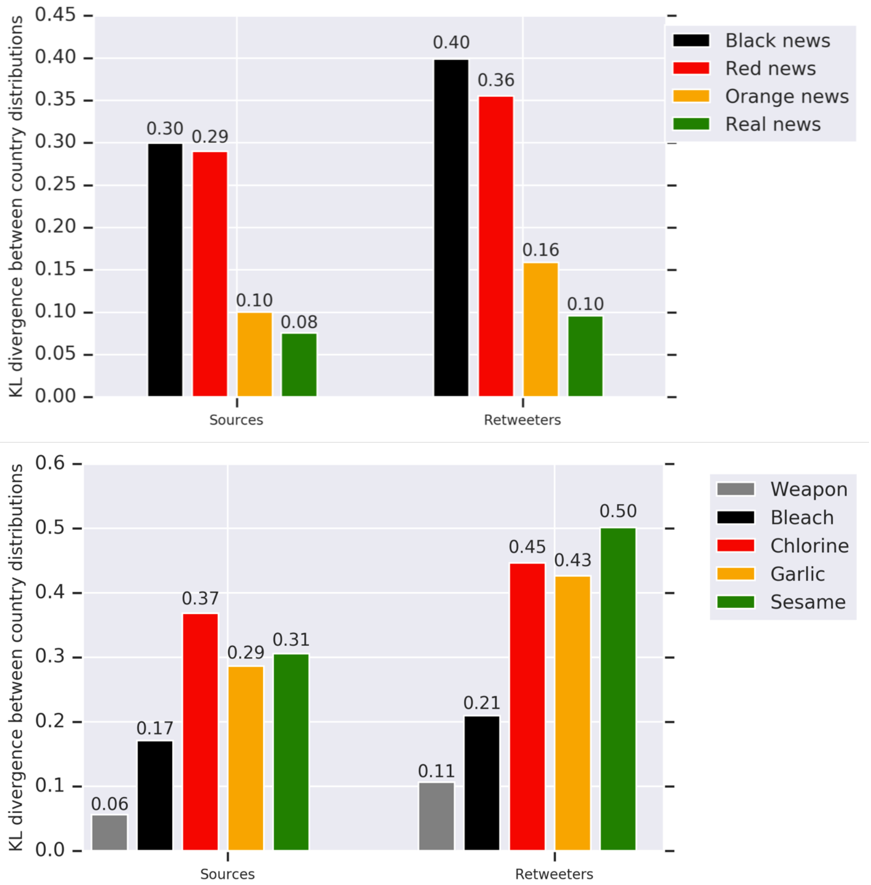


Fig. 6.5 KL-divergence between country distributions of EN users sharing news and all EN users (above). KL-divergence between country distributions of EN users talking misinformation stories and all EN users (bottom).

with “fake news” URLs and misinformation conversations. As shown in the above figure, 39.31% of users who retweet tweets with real news URLs are from a different country than the source tweet. The percentage of inter-country retweets is much lower for the tweets containing URLs to less credible sites, especially for the black news sites. This demonstrates that tweets mentioning less credible news-sites tend to stay within the source country. This helps explain why country distribution of users sharing these “fake news” URLs differ from the underlying population.

As for the misinformation stories, tweets talking about bio-weapon, bleach, chlorine dioxide are more likely to be retweeted by users from the same country. Tweets mentioning garlic and sesame are more likely to spread internationally. One reason for this is that global health agencies such as WHO posted several clarifications for these misinformation stories.

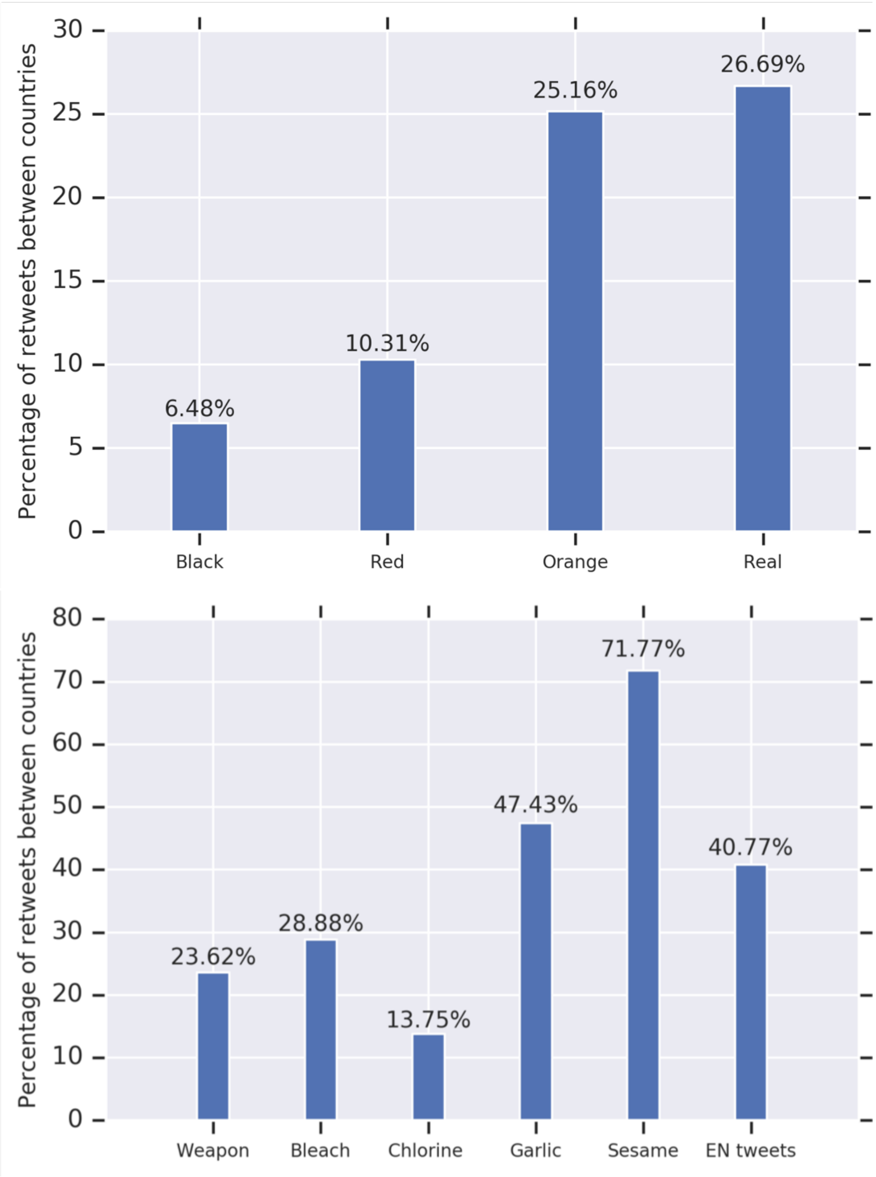


Fig. 6.6 Percentages of retweets between countries for "fake news" sites and misinformation conversations.

As a result, conversations about real news and certain misinformation stories have high country diversity in their spread. Here, we use entropy of country distribution to measure the country diversity of Twitter users who shared each "fake news" sites and misinformation stories. As shown in Figure 6.7, the country entropy of users who posted tweets with "fake news" URLs are much less than users who posted real news URLs. The black news and red news are constrained in the source countries. Similar effect also happens for misinformation stories. Tweets talking about garlic and sesame are spreading as diverse as normal tweets, while users talking about chlorine dioxide are highly concentrated in certain countries.

In Figure 6.8, we show the information flows among countries. The width of a flow is proportional to the percentage of retweets from country A to country B among retweets between all the countries. We only show information flows with percentages higher than 0.5%. We also exclude the retweets inside the same country in this figure. As shown in Figure 6.8a, United States contributing the most of the retweeting flow between countries. Out of 11,092,477 inter-country retweets, 4,034,985 are from United States (36.38%). A great deal of information moves from the United States to other countries particularly the United Kingdom and Hong Kong. The situations become more extreme in the cases of bio-weapon (39,535 out of 69,389, 56.98%), bleach (5,803 out of 8,144, 71.25%), and chlorine dioxide (292 out of 326, 89.6%). Switzerland, where WHO is located, plays a larger role in the flow of information about sesame and garlic. For example, for the sesame story-line there are 823 retweets between countries, 415 of them are from Switzerland (50.43%). All of these retweets are people retweeting WHO.

6.4 Discussion

This study investigated the discussion about the novel coronavirus on Twitter. We examined fake news URLs and misinformation stories spreading in this emergence event. Our study shows that news agencies, government officials, and individual news reporters do send messages that spread widely, and so play critical roles. However, the most influential tweets are those posted by regular users, some of whom are bots. Tweets mentioning fake news URLs and misinformation stories are more likely to be spread by regular users than the news or government accounts. The distribution of users mentioning the URLs of less credible news sites across countries is different from the distribution of users mentioning real news URLs. More users mentioning these less credible sites and/or the disinformation story-lines come from United States. Unlike messages that mention real news URLs or don't discuss these disinformation story lines which often spread between countries, these "fake news" discussions typically spread within a country.

In this paper, we utilized machine learning systems to predict users' latent attributes, such as their locations and political orientations. Even though our prediction systems have reasonably high accuracy, they are still prone to prediction errors for individual users. To ensure the reliability of our analysis, we have focused on aggregated results. For the same reason we focused on comparative results instead of absolute values;

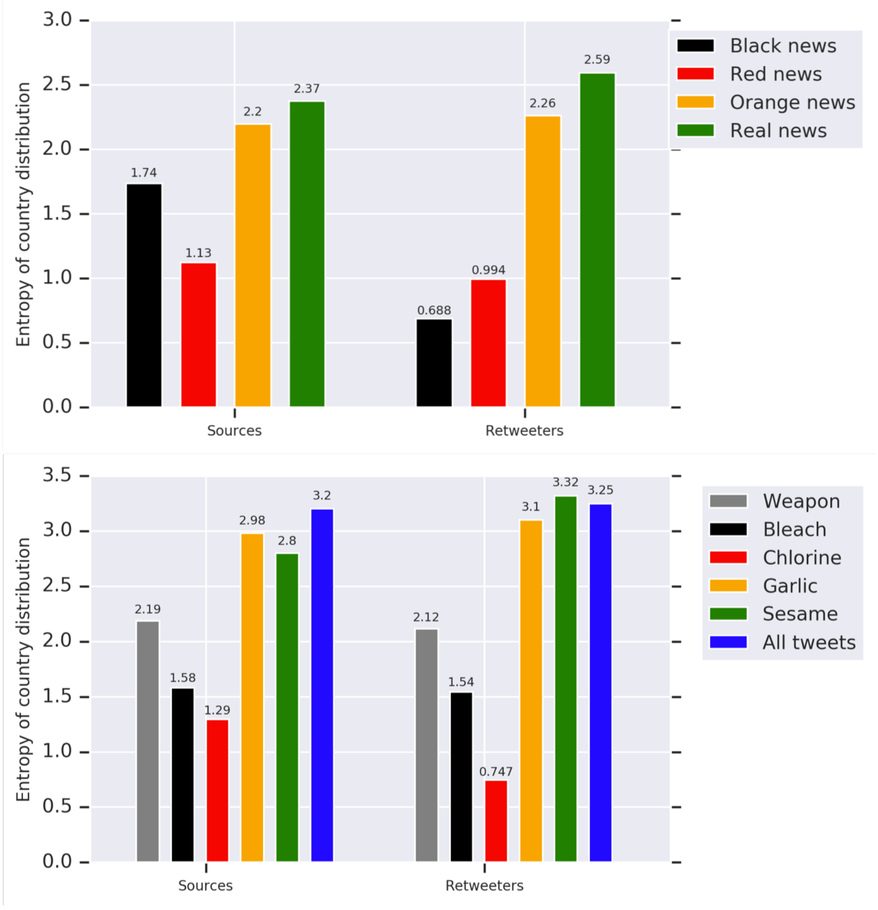
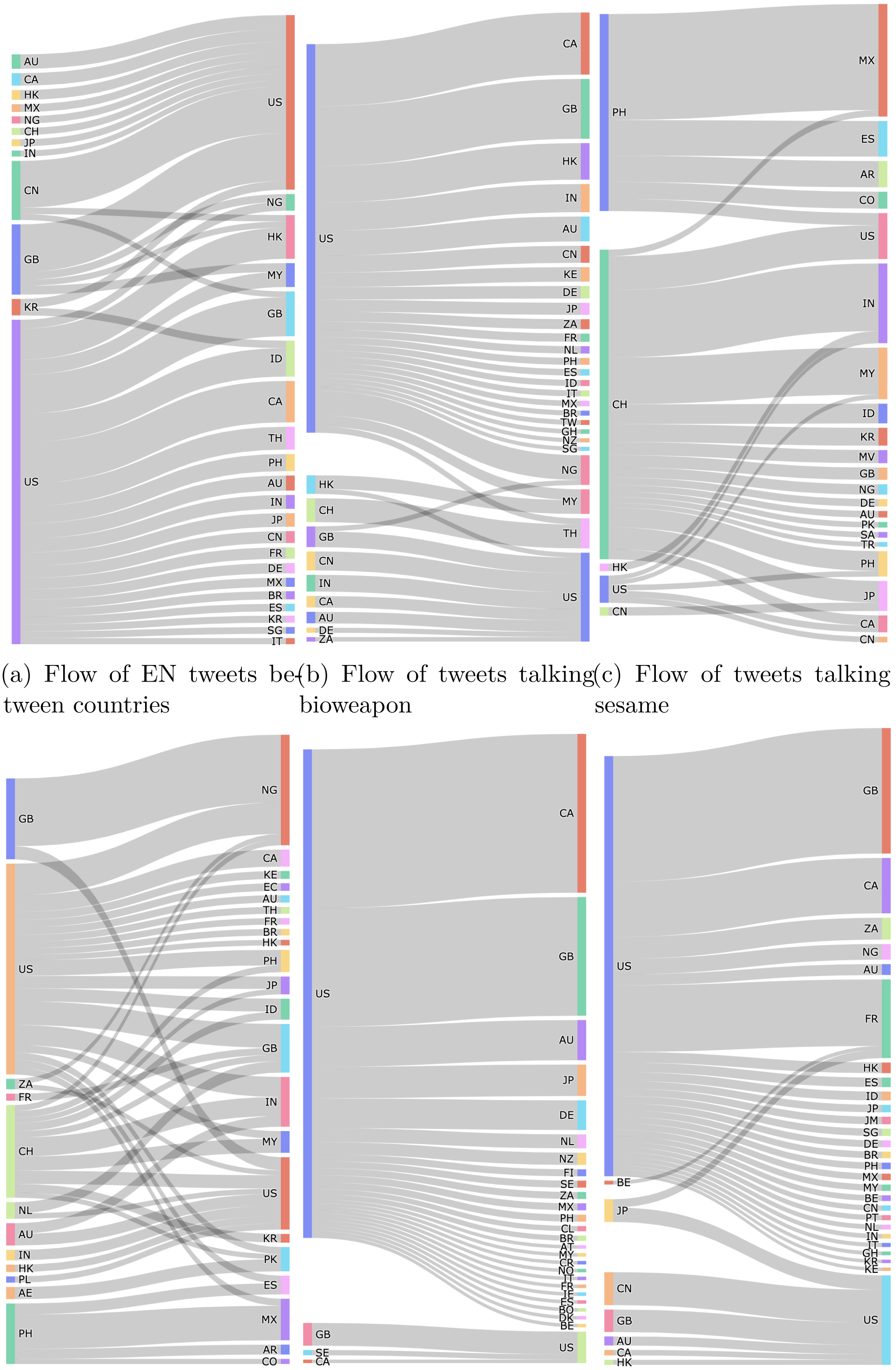


Fig. 6.7 Entropy of country distribution for users who posted certain "fake news" URLs and misinformation stories.

e.g., there is a higher percentage of conservative users involved in the bio-weapon conspiracy tweets than in non-conspiracy tweets. We also tested the generalizability of the machine learning systems on this COVID-19 dataset. For the location prediction system, we re-trained it using the geotagged users in this dataset. For the identity and political orientation prediction models, we extracted all test users existing in the COVID-19 dataset and re-collected their most recent 200 tweets in this time period. The testing accuracies for this subset of test users are 90.2% and 91.4% which are very close to the performance 90.6% and 90.6% on identity and political orientation classification respectively evaluated on the original dataset. In order to get a consistent evaluation on the political ideology of global users, we apply a political orientation prediction system trained on users from the United States on English speaking users globally. The political orientation prediction results should be interpreted relative to the current conservative versus liberal differences in the United States. We note that, to first order, these same differences are prevalent in Western Europe.

It is important to note that this study only extracted tweets containing certain types of news URLs or certain keywords associated with disinformation story-lines. Whether these tweets are being spread by those knowing they are inaccurate maliciously, as a joke, or simply to discuss the inaccuracy is not considered in this paper. Future work, separating these tweets by the original purpose could provide us with a better understanding of how disinformation spreads during an emergency and the conditions under which it needs to be countered. Our search criteria for finding tweets coronavirus resulted in a bias toward tweets in the English language. Our search was also constrained by the limits of the Twitter APIs. Hence, there are likely to be additional conversations related to this pandemic that are not captured. Future work might consider automatically detecting new tracking keywords in the streaming data to dynamically shift the selection and so capture more conversations as the conversation drifts between topics [66].

There are many potential implications of this study. We found that regular users sent the majority of tweets referring to non-credible news sites and mentioning the disinformation story-lines. Many of these regular users appear to be bots; however, most people cannot recognize bots. Thus, people should be cautious when reading tweets sent from regular users, and perhaps be even more skeptical when reading those posts, than those from news agencies and the government. Regular users in some countries appear to be greater consumers of information and sources lacking credibility. This suggests that local country regulations, policies, and technology may be important in reducing the spread of such information. We found that health authorities, such as WHO, played a critical role. Concerted efforts to increase the reach of such authorities may be of value in combating misinformation.



(a) Flow of EN tweets between countries

(b) Flow of tweets talking bioweapon

(c) Flow of tweets talking sesame

(d) Flow of tweets talking garlic

(e) Flow of tweets talking chlorine dioxide

(f) Flow of tweets talking bleach

Fig. 6.8 Information flows among countries. Retweets from the same country as the source are excluded in this figure. We use ISO 3166-1 alpha-2 country code.

Chapter 7

Conclusions and Discussions

7.1 Summary of Contributions

This thesis aims to learn user latent attributes from their social media posts. These attributes such as location and politic orientation are often not explicitly available from user's homepage and some implicit indicative information are often buried in the large volume of noisy social media content, which makes this task challenging. In this thesis, I uncover these implicit indicative information with machine learning. I propose to learn user latent attributes from three levels – tweet-level, user-level, and graph-level. The proposed methods utilize heterogeneous data available on social media including tweet text, user profile information, as well as social networks.

I start the thesis with a simple tweet-level tweet-level location prediction system. It is able to geolocate one user based on the information in a single tweet object. The proposed approach integrates tweet text and user profile meta-data into a single model. Compared to the previous stacking method with feature selection, our approach substantially outperforms the baseline method. We developed the approach for both city and country level and demonstrated the ability to classify almost 50% and 90% of all tweets at city-level and country-level respectively.

I further improve the performance of the tweet-level system by incorporating more information from user's whole timeline. We propose a hierarchical self-attention neural network to learn useful features from multiple tweets for each user. Our experiments demonstrate that the proposed model significantly outperforms multiple baselines. Based on this user-level model, we propose a hierarchical location prediction neural network to improve Twitter user geolocation. It first predicts the home country for a user, then uses the country result to guide the city-level prediction. It not only improves the prediction accuracy but also greatly reduces the mean error distance.

All the previous presented methods ignore the social network factor, which may provide additional information to learn user's attributes because of social homophily effect. In the fifth chapter, I propose a unified model which combines my user-level model and a graph neural network into one single prediction model. I demonstrate that we can get much better performance when considering network information. Furthermore, I also show that classic network metrics could complement the graph neural network by adding network structure information.

Given all the tools developed in this thesis, I investigated the discussion about the novel coronavirus on Twitter as a case study. My studies show that news reporters, government officials, and individual news reporters play an important role in this event. Generally, tweets written by these users are more likely to spread widely. However, the most influential tweets are still posted by regular users. In the mean while, tweets with fake news URLs and misinformation stories are also more likely to be spread by regular users. In this particular event, fake news country distribution differs from real news spreading pattern. A higher proportion of tweets talking about fake news and misinformation come from United States. Besides, unlike real news and normal tweets, tweets with fake news URLs are constrained inside the source country and are less likely to spread internationally. As a result, tweets with fake news URLs spread to countries with lower diversity than tweets with real news URLs. We have similar observation for tweets talking about certain misinformation stories.

Because it is relatively expensive to get manually labeled data, in this thesis I also explored various ways to use unlabeled users to enhance the performance. In the fourth chapter, my experiments show that we can utilize the Twitter's verify field as a noisy label to learn identity features of public figures. The knowledge learned from public figures can be transferred to the more fine-grained identity classification task, which reduces the need of manual labeling and improves the performance. In the fifth chapter, I also show that in the semi-supervised setting we can greatly improve the classification performance by adding unlabeled users.

Last, I want to note that though I mainly use tweet data to predict user attributes like location, social identity, the methodology can be easily extended to other platforms like Facebook and Weibo, as well as other characteristics, eg. gender and age.

7.2 Limitations & Future Work

There are some limitations of the work presented in this thesis. I list several major ones as follows.

For these latent attributes, our methods assume they are static through time. However, some of these attributes may evolve overtime when users change their statuses. For example, people may relocate to other cities because of traveling. However, our model assumes each post of one user all comes from one single home location but ignores the dynamic user movement pattern. For other attributes such as political orientation, people may also change their ideology slowly overtime. We plan to incorporate temporal states to capture attributes changes in future work.

Besides, in this thesis we only considered limited dimensions of latent attributes. However, attributes such as political orientation may have various dimensions. Users we labeled as liberal may be in favor of certain political policies of the democratic party while do not support the other democratican policies. Also, users in other foreign countries may share a different politic ideology system. Users considered to be mild conservative in one country may be considered as liberal in another country. Similar for identity classification, in the real-world people often have multiple identities - e.g., Serbian, Entrepreneur, Policewoman, Woman, Mother. In social media, some people use different accounts and/or different social media platforms for different identities - e.g., Facebook for Mother, Twitter for Entrepreneur and a separate Twitter handle for official policewoman account. In this paper, we made no effort to determine whether an individual had multiple accounts. Thus, the same user may get multiple classifications if that user has multiple accounts. Future work should explore how to link multiple identities to the same user.

For future work, we may also consider increase the granularity of our prediction labels. In the case of city-level location prediction, we have over 3000 major populated cities around world. If we want to increase the granularity, we may consider divide earth into tens of thousands grids. Then, we place each user into these fine-grained grids. However, when the number of classes increases to hundreds of thousands, we may not have enough training data for every target class. Alternatively, we may also consider convert the problem into coordinates regression that learn the precise coordinates of users. However, such precise geolocation would put individual's privacy at risk. In the case of identity classification, currently we do not have a hierarchical ontology of identities. For future work, we also need to design such identity ontology and predict users' identities at different levels, which also enables multiple identities to be assigned to the same account and so to the same user. For political orientation prediction, in reality people may have different political opinion towards certain topics. In the future, we plan to tackle this issue by looking at aspect-specific political orientation prediction. Hence, we can get users' stances of different issues. Given a pair of input —

user's profile and a political issue, we can output a stance of each user for this specific issue. In the meanwhile, we also need to control the granularity of our prediction so that individual users cannot be separated out from a group of users due to privacy consideration.

Currently, our methods are trained on previously collected datasets. However, topics discussed on social media evolves very quickly. People may still discuss the superbowl in early February 2020, while the major topic changes to the novel coronavirus very soon. The performance of machine learning models trained on data collected years before may not generalize to the current data pattern. To accommodate the newly incoming data stream, we may consider a never-ending learning paradigm [76].

In addition to the timing effect, attributes prediction models trained on a limited number of training data may also suffer from the domain bias. When we apply our prediction systems in the wild, it is still questionable that can our models generalize to users in another domain. Using users' whole timeline and social links may minimize the bias effect, since their long tweeting histories and social ties are less prone to the topic drift. When apply our models on a new dataset, one can also try the same method as what we did in the Chapter 6 that test the performance on a subset of overlapped users. If the performance drops significantly, we may also consider unsupervised domain adaption technique [38] to better adapt our models in a new domain.

Previous study also shows that deep neural networks can be easily fooled by some simple data manipulation [103]. If malicious users such as social bots want to hide their identities or pretend themselves to be someone else, they may intentionally manipulate their social posts and fool such learning algorithms. In the future work, we would like to consider a more robust neural network model.

Last but not least, though this work proposes multiple ways to reduce the need of labeled users such as transfer learning and semi-supervised learning, it still requires human supervision to learn user representations, which is very expensive. One possible way to learn universal representations without labeled data is via self-supervision [33]. BERT already demonstrates we can learn a powerful language model which could be adapted to multiple sub-tasks. Since we not only have text, but also metadata and graphs, it is still challenging to learn a universal user-level model that takes the heterogeneous data sources into consideration.

7.3 Privacy Considerations

One concern of studying user behavior on social media is privacy. By default tweets posted by users are considered as public information and anyone on Internet can access public tweets without restriction[1]. There are an option for users that they can choose to hide their profiles and unrelated persons are not allowed to view their profiles. In this paper, we only collected data that are public available following users' privacy setting. The collected datasets are only used for research and are not shared with any third-parties. In the meanwhile, the trained models are also used internally for research purpose.

This thesis only presents results for groups of users on an aggregated level. The data reported in this paper does not involve any individual's sensitive information. This thesis demonstrates that we can combine conversations on social media and machine learning techniques during a global disaster for social good.

[1] https://twitter.com/en/privacy

References

[1] Achrekar, H., Gandhe, A., Lazarus, R., Yu, S.-H., and Liu, B. (2011). Predicting flu trends using twitter data. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 702–707. IEEE.

[2] Al Zamal, F., Liu, W., and Ruths, D. (2012). Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors. In *Sixth International AAAI Conference on Weblogs and Social Media*.

[3] Allcott, H. and Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36.

[4] Amitay, E., Har'El, N., Sivan, R., and Soffer, A. (2004). Web-a-where: geotagging web content. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 273–280. ACM.

[5] Ashforth, B. E. and Mael, F. (1989). Social identity theory and the organization. *Academy of management review*, 14(1):20–39.

[6] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

[7] Babcock, M., Beskow, D. M., and Carley, K. M. (2019). Different faces of false: The spread and curtailment of false information in the black panther twitter discussion. *Journal of Data and Information Quality (JDIQ)*, 11(4):1–15.

[8] Babcock, M., Villa-Cox, R., and Carley, K. M. (2020). Pretending positive, pushing false: Comparing captain marvel misinformation campaigns. *Disinformation, Misinformation, and Fake News in Social Media-Emerging Research Challenges and Opportunities*.

[9] Backstrom, L., Sun, E., and Marlow, C. (2010). Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th international conference on World wide web*, pages 61–70. ACM.

[10] Berggren, M., Karlgren, J., Östling, R., and Parkvall, M. (2016). Inferring the location of authors from words in their texts. *arXiv preprint arXiv:1612.06671*.

[11] Beskow, D. and Carley, K. M. (2020a). *Social Cybersecurity*. Springer.

[12] Beskow, D. and Carley, K. M. (2020b). You are known by your friends: Leveraging network metrics for bot detection in twitter. *Open Source Intelligence and Cyber Crime*.

[13] Beskow, D., Carley, K. M., Bisgin, H., Hyder, A., Dancy, C., and Thomson, R. (2018). Introducing bothunter: A tiered approach to detection and characterizing automated activity on twitter. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation. Springer.*

[14] Bessi, A., Zollo, F., Del Vicario, M., Scala, A., Caldarelli, G., and Quattrociocchi, W. (2015). Trend of narratives in the age of misinformation. *PloS one*, 10(8).

[15] Bilhaut, F., Charnois, T., Enjalbert, P., and Mathet, Y. (2003). Geographic reference analysis for geographic document querying. In *Proceedings of the HLT-NAACL 2003 workshop on Analysis of geographic references-Volume 1*, pages 55–62. Association for Computational Linguistics.

[16] Bo, H., Cook, P., and Baldwin, T. (2012). Geolocation prediction in social media data by finding location indicative words. In *Proceedings of COLING*, pages 1045–1062.

[17] Bovet, A. and Makse, H. A. (2019). Influence of fake news in twitter during the 2016 us presidential election. *Nature communications*, 10(1):1–14.

[18] Burger, J. D., Henderson, J., Kim, G., and Zarrella, G. (2011). Discriminating gender on twitter. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1301–1309. Association for Computational Linguistics.

[19] Buyukokkten, O., Cho, J., Garcia-Molina, H., Gravano, L., and Shivakumar, N. (1999). Exploiting geographical location information of web pages.

[20] Calhoun, C. J. (1994). Social theory and the politics of identity.

[21] Callero, P. L. (1985). Role-identity salience. *Social psychology quarterly*, pages 203–215.

[22] Carley, K. M., Malik, M., Landwehr, P. M., Pfeffer, J., and Kowalchuck, M. (2016a). Crowd sourcing disaster management: The complex nature of twitter usage in padang indonesia. *Safety science*, 90:48–61.

[23] Carley, K. M., Wei, W., and Joseph, K. (2016b). High-dimensional network analytics: mapping topic networks in twitter data during the arab spring.

[24] Cha, M., Haddadi, H., Benevenuto, F., Gummadi, P. K., et al. (2010). Measuring user influence in twitter: The million follower fallacy. *Icwsm*, 10(10-17):30.

[25] Chatfield, A. T. and Brajawidagda, U. (2013). Twitter early tsunami warning system: A case study in indonesia's natural disaster management. In *2013 46th Hawaii International Conference on System Sciences*, pages 2050–2060. IEEE.

[26] Cheng, Z., Caverlee, J., and Lee, K. (2010). You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 759–768. ACM.

[27] Chu, Z., Gianvecchio, S., Wang, H., and Jajodia, S. (2010). Who is tweeting on twitter: human, bot, or cyborg? In *Proceedings of the 26th annual computer security applications conference*, pages 21–30. ACM.

[28] Chu, Z., Gianvecchio, S., Wang, H., and Jajodia, S. (2012). Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Transactions on Dependable and Secure Computing*, 9(6):811–824.

[29] Colleoni, E., Rozza, A., and Arvidsson, A. (2014). Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. *Journal of Communication*, 64(2):317–332.

[30] Conneau, A., Schwenk, H., Barrault, L., and Lecun, Y. (2016). Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*.

[31] Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. (2017). Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941. JMLR. org.

[32] Del Vicario, M., Bessi, A., Zollo, F., Petroni, F., Scala, A., Caldarelli, G., Stanley, H. E., and Quattrociocchi, W. (2016). The spreading of misinformation online. *Proceedings of the National Academy of Sciences*, 113(3):554–559.

[33] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[34] Do, T. H., Nguyen, D. M., Tsiligianni, E., Cornelis, B., and Deligiannis, N. (2017). Multiview deep learning for predicting twitter users' location. *arXiv preprint arXiv:1712.08091*.

[35] Donzelli, G., Palomba, G., Federigi, I., Aquino, F., Cioni, L., Verani, M., Carducci, A., and Lopalco, P. (2018). Misinformation on vaccination: A quantitative analysis of youtube videos. *Human vaccines & immunotherapeutics*, 14(7):1654–1659.

[36] Earle, P. S., Bowden, D. C., and Guy, M. (2012). Twitter earthquake detection: earthquake monitoring in a social world. *Annals of Geophysics*, 54(6).

[37] Ebrahimi, M., ShafieiBavani, E., Wong, R., and Chen, F. (2018). A unified neural network model for geolocating twitter users. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 42–53.

[38] Ganin, Y. and Lempitsky, V. (2014). Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*.

[39] Godoy, D. and Amandi, A. (2005). User profiling for web page filtering. *IEEE Internet computing*, 9(4):56–64.

[40] Graham, M., Hale, S. A., and Gaffney, D. (2014). Where in the world are you? geolocation and language identification in twitter. *The Professional Geographer*, 66(4):568–578.

[41] Grinberg, N., Joseph, K., Friedland, L., Swire-Thompson, B., and Lazer, D. (2019). Fake news on twitter during the 2016 us presidential election. *Science*, 363(6425):374–378.

[42] Gupta, A., Lamba, H., Kumaraguru, P., and Joshi, A. (2013). Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd international conference on World Wide Web*, pages 729–736.

[43] Hale, S., Gaffney, D., and Graham, M. (2012). Where in the world are you? geolocation and language identification in twitter. *Proceedings of ICWSM*, 12:518–521.

[44] Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034.

[45] Han, B., Cook, P., and Baldwin, T. (2012). Geolocation prediction in social media data by finding location indicative words. *Proceedings of COLING 2012*, pages 1045–1062.

[46] Han, B., Cook, P., and Baldwin, T. (2013). A stacking-based approach to twitter user geolocation prediction. In *ACL (Conference System Demonstrations)*, pages 7–12.

[47] Han, B., Cook, P., and Baldwin, T. (2014). Text-based twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, 49:451–500.

[48] Han, B., Rahimi, A., Derczynski, L., and Baldwin, T. (2016). Twitter geolocation prediction shared task of the 2016 workshop on noisy user-generated text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 213–217.

[49] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[50] Hecht, B., Hong, L., Suh, B., and Chi, E. H. (2011). Tweets from justin bieber's heart: the dynamics of the location field in user profiles. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 237–246. ACM.

[51] Hentschel, M., Alonso, O., Counts, S., and Kandylas, V. (2014). Finding users we trust: Scaling up verified twitter users using their communication patterns. In *ICWSM*.

[52] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[53] Hong, L., Ahmed, A., Gurumurthy, S., Smola, A. J., and Tsioutsiouliklis, K. (2012). Discovering geographical topics in the twitter stream. In *Proceedings of the 21st international conference on World Wide Web*, pages 769–778. ACM.

[54] Huang, B. and Carley, K. M. (2017). On predicting geolocation of tweets using convolutional neural networks. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 281–291. Springer.

[55] Huang, B. and Carley, K. M. (2019a). A hierarchical location prediction neural network for twitter user geolocation. *EMNLP*.

[56] Huang, B. and Carley, K. M. (2019b). A large-scale empirical study of geotagging behavior on twitter. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '19, page 365–373, New York, NY, USA. Association for Computing Machinery.

[57] Huang, B. and Carley, K. M. (2019c). Residual or gate? towards deeper graph neural networks for inductive graph representation learning. *arXiv preprint arXiv:1904.08035*.

[58] Huang, B. and Carley, K. M. (2020). Discover your social identity from what you tweet: a content based approach. *Disinformation, Misinformation, and Fake News in Social Media - Emerging Research Challenges and Opportunities*.

[59] Jayasinghe, G., Jin, B., Mchugh, J., Robinson, B., and Wan, S. (2016). Csiro data61 at the wnut geo shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 218–226.

[60] Jenkins, R. (2014). *Social identity*. Routledge.

[61] Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

[62] Jurgens, D. (2013). That's what friends are for: Inferring location in online social media platforms based on social relationships. *ICWSM*, 13:273–282.

[63] Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[64] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[65] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

[66] Kumar, S. and Carley, K. M. (2019). What to track on the twitter streaming api? a knapsack bandits approach to dynamically update the search terms. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 158–163.

[67] Landwehr, P. M. and Carley, K. M. (2014). Social media in disaster relief. In *Data mining and knowledge discovery for big data*, pages 225–257. Springer.

[68] Lazer, D. M., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., et al. (2018). The science of fake news. *Science*, 359(6380):1094–1096.

[69] Lee, N. (2014). Facebook nation. *Total information awareness. Nueva York: Springer.*

[70] Loeb, S., Sengupta, S., Butaney, M., Macaluso Jr, J. N., Czarniecki, S. W., Robbins, R., Braithwaite, R. S., Gao, L., Byrne, N., Walter, D., et al. (2019). Dissemination of misinformative and biased information about prostate cancer on youtube. *European urology*, 75(4):564–567.

[71] Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3.

[72] Mahmud, J., Nichols, J., and Drews, C. (2012). Where is this tweet from? inferring home locations of twitter users. In *Sixth International AAAI Conference on Weblogs and Social Media.*

[73] McPherson, M., Smith-Lovin, L., and Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444.

[74] Middleton, S. E., Shadbolt, N. R., and De Roure, D. C. (2004). Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):54–88.

[75] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[76] Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Yang, B., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., et al. (2018). Never-ending learning. *Communications of the ACM*, 61(5):103–115.

[77] Miura, Y., Taniguchi, M., Taniguchi, T., and Ohkuma, T. (2016). A simple scalable neural networks based model for geolocation prediction in twitter. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 235–239.

[78] Miura, Y., Taniguchi, M., Taniguchi, T., and Ohkuma, T. (2017). Unifying text, metadata, and user network representations with a neural network for geolocation prediction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1260–1272.

[79] Miyazaki, T., Rahimi, A., Cohn, T., and Baldwin, T. (2018). Twitter geolocation using knowledge-based methods. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 7–16.

[80] Mocanu, D., Rossi, L., Zhang, Q., Karsai, M., and Quattrociocchi, W. (2015). Collective attention in the age of (mis) information. *Computers in Human Behavior*, 51:1198–1204.

[81] Overell, S. E. (2009). *Geographic information retrieval: Classification, disambiguation and modelling.* PhD thesis, Citeseer.

[82] Pennacchiotti, M. and Popescu, A.-M. (2011). A machine learning approach to twitter user classification. In *Fifth International AAAI Conference on Weblogs and Social Media*.

[83] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[84] Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM.

[85] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

[86] Priante, A., Hiemstra, D., van den Broek, T., Saeed, A., Ehrenhard, M., and Need, A. (2016). # whoami in 160 characters? classifying social identities based on twitter profile descriptions. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 55–65.

[87] Qi, J., Trang, T., Doong, J., Kang, S., and Chien, A. L. (2016). Misinformation is prevalent in psoriasis-related youtube videos. *Dermatology online journal*, 22(11).

[88] Qian, Y., Tang, J., Yang, Z., Huang, B., Wei, W., and Carley, K. M. (2017). A probabilistic framework for location inference from social media. *arXiv preprint arXiv:1702.07281*.

[89] Quercia, D., Lathia, N., Calabrese, F., Di Lorenzo, G., and Crowcroft, J. (2010). Recommending social events from mobile phone location data. In *2010 IEEE international conference on data mining*, pages 971–976. IEEE.

[90] Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

[91] Rahimi, A., Cohn, T., and Baldwin, T. (2015a). Twitter user geolocation using a unified text and network prediction model. *arXiv preprint arXiv:1506.08259*.

[92] Rahimi, A., Cohn, T., and Baldwin, T. (2017). A neural model for user geolocation and lexical dialectology. *arXiv preprint arXiv:1704.04008*.

[93] Rahimi, A., Cohn, T., and Baldwin, T. (2018). Semi-supervised user geolocation via graph convolutional networks. *arXiv preprint arXiv:1804.08049*.

[94] Rahimi, A., Vu, D., Cohn, T., and Baldwin, T. (2015b). Exploiting text and network context for geolocation of social media users. *arXiv preprint arXiv:1506.04803*.

[95] Rao, D., Yarowsky, D., Shreevats, A., and Gupta, M. (2010). Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 37–44. ACM.

[96] Recasens, M., Hovy, E., and Martí, M. A. (2011). Identity, non-identity, and near-identity: Addressing the complexity of coreference. *Lingua*, 121(6):1138–1152.

[97] Robinson, L. (2007). The cyberself: the self-ing project goes online, symbolic interaction in the digital age. *New Media & Society*, 9(1):93–110.

[98] Roller, S., Speriosu, M., Rallapalli, S., Wing, B., and Baldridge, J. (2012). Supervised text-based geolocation using language models on an adaptive grid. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1500–1510. Association for Computational Linguistics.

[99] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

[100] Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM.

[101] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

[102] Shao, C., Hui, P.-M., Wang, L., Jiang, X., Flammini, A., Menczer, F., and Ciampaglia, G. L. (2018). Anatomy of an online misinformation network. *PloS one*, 13(4):e0196087.

[103] Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K. (2016). Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540.

[104] Shmueli-Scheuer, M., Roitman, H., Carmel, D., Mass, Y., and Konopnicki, D. (2010). Extracting user profiles from large scale data. In *Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud*, page 4. ACM.

[105] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

[106] Stryker, S. and Burke, P. J. (2000). The past, present, and future of an identity theory. *Social psychology quarterly*, pages 284–297.

[107] Tajfel, H. (1974). Social identity and intergroup behaviour. *Information (International Social Science Council)*, 13(2):65–93.

[108] Tajfel, H. (1982). *Social identity and intergroup relations.* Cambridge University Press.

[109] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077. International World Wide Web Conferences Steering Committee.

[110] Tsou, M.-H., Yang, J.-A., Lusher, D., Han, S., Spitzberg, B., Gawron, J. M., Gupta, D., and An, L. (2013). Mapping social activities and concepts with social media (twitter) and web search engines (yahoo and bing): a case study in 2012 us presidential election. *Cartography and Geographic Information Science*, 40(4):337–348.

[111] Uyheng, J. and Carley, K. M. (2019). Characterizing bot networks on twitter: An empirical analysis of contentious issues in the asia-pacific. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 153–162. Springer.

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

[113] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

[114] Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. *Science*, 359(6380):1146–1151.

[115] Wing, B. and Baldridge, J. (2014). Hierarchical discriminative classification for text-based geolocation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 336–348.

[116] Wing, B. P. and Baldridge, J. (2011). Simple supervised document geolocation with geodesic grids. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 955–964. Association for Computational Linguistics.

[117] Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.

[118] Zhang, Y., Wei, W., Huang, B., Carley, K. M., and Zhang, Y. (2017). Rate: Overcoming noise and sparsity of textual features in real-time location estimation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2423–2426. ACM.

Appendix A

Data Collections

Tweet-level location dataset

We used geo-tagged tweets collected from Twitter streaming API[1]. We set the geographic bounding box as [-180, -90, 180, 90] so that we could get these geo-tagged tweets from the whole world. Our collection started from January 7, 2017 to February 1, 2017. The dataset is stored at /storage2/binxuan/data/tweet_global_location on CASOS's ECE servers.

We are only using tweets either with specific geo-coordinates or a geo-bounding box smaller than [0.1,0.1]. The country label is directly retrieved from geotags in tweets. The city label is created by assigning tweets to the major populated cities. The city list is in the file of city_collapsed_all.

There are 3,321,194 users and 4,645,692 tweets in total. We randomly selected 10% users' tweets as testing data. For the remaining 90% users, we picked tweets from 50,000 of them as a development set and used the remaining tweets as training data. The training data is saved in 5 files with name starting with "training_". The test data is saved in 2 files with name starting with "testing_". The "dev_0" contains development data. Each line of these files represents one single tweet. Each line is organized as

country_index city_index tweet_text description location tweet_lang_index user_lang_index timezone_index time_index.

[1] https://developer.twitter.com/en/docs/tweets/filter-realtime/api-reference/post-statuses-filter

You can find the name to index mapping in files “langMap”, “countryMap”, “timezoneMap” for language, country, and timezone respectively. The city index is the line number for each city in the file city_collapsed_all starting from 0.

Public Figure

We use Twitter’s verification as a proxy for public figures, and these verified accounts include users in music, government, sports, business, and etc[2]. We sampled 156746 verified accounts and 376371 unverified accounts through Twitter’s sample stream data [3]. Then we collected their most recent 200 tweets from Twitter’s API[4] in November 2018. We randomly choose 5000 users as a development set and 10000 users as a test set.

The dataset is saved at /storage2/binxuan/data/verify_full on CASOS’s ECE servers. Files with name “output_*_.json.gz” contains the raw collected tweet JSONs. After de-compressing, each line of these file represents one single tweet JSON. The user label is contained in the “verified” field in the JSON. The file following_reduced.csv contains the following ties among these users. Each line is organized as “source, target (followee)”. It is collected from Twitter’s API[5].

preprocess_v2.ipynb is a jupyter notebook that runs a spark application to preprocess the raw json files. “user.json” is the concatenated output file which contains user-based data point. Each line is a user JSON object with fields “user_id”, “description”, “name”, “screen_name”, “label”, and “tweets”.

Identity

The identity dataset is saved at /storage2/binxuan/data/identity2 on CASOS’s ECE servers. In the “data” folder, each csv file contains the user IDs for each corresponding identity category. In the same folder, we also have the corresponding raw json files containing the most recent 200 tweets collected using API[6].

[2] https://help.twitter.com/en/managing-your-account/about-twitter-verified-accounts

[3] https://developer.twitter.com/en/docs/tweets/sample-realtime/overview/GET_statuse_sample.html

[4] https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user$_timeline$

[5] https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friends-ids

[6] https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user$_timeline$

preprocess_v2.ipynb is a jupyter notebook that runs a spark application to preprocess the raw json files. “user.json” is the concatenated output file which contains user-based data point. Each line is a user JSON object with fields “user_id”, “description”, “name”, “screen_name”, “label”, and “tweets”.

Political Orientation

The political orientation dataset is saved at /storage2/binxuan/data/politic_final on CASOS’s ECE servers. The seed politicians’ accounts are included in the file “political_affialiation.csv”. Each line of it contains the screen name, name, political party, and position of each politicians. followers_parties.csv is the edgelist file that contains the followers of these politicians belonging to democratic or republican. We sampled a subset of users only following politicians of one party and saved them at collect_final.csv. Each line of this file is “follower_id, party_index”. Party index “1” represents conservative.

We collected the timelines of these users using API[7]. The folder “timeline” contains raw JSON files of the most recent 200 tweets for all the users. following_reduced.csv contains the following ties among these users. Each line is organized as “source, target (followee)”. It is collected from Twitter’s API[8].

Coronavirus

The coronavirus dataset is created by searching coronavirus related keywords in the real-time Twitter stream starting from Jan. 29, 2020. The initial keywords set contains “coronaravirus”, “coronavirus”, “wuhan virus”, “wuhanvirus”. Then we added keywords “NCoV” on Jan. 30 and “covid-19”, “covid19”, “covid 19” on March 11. These files saved are saved at /storage3/coronavirus/json_keyword_stream/ on CASOS’s ECE servers. The file names are “virus_year_month_day.json.gz”.

The timeline data is saved at /storage3/binxuan/data/virus with name pattern “timeline_start_date_end_date”. Each folder contains the most recent 200 tweets for users who have only posted after the start date and before the end date (new unique users appearing after the start date.). The following edgelists are also saved at /storage3/binxuan/data/virus with name pattern “following_start_date_end_date.csv”.

[7] https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user$_{t}imeline$

[8] https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friends-ids

Each folder contains following ties for users who have only posted after the start date and before the end date.