Technical Report

Predicting User Behaviour at Twitter

Predicting User Behaviour at Twitter

Daniel Wang
University of Virginia
Charlottesville, VA, USA
hw4ce@virginia.edu

Mandy Wilson
BioComplexity Institute,
University of Virginia
Charlottesville, VA, USA
alw4ey@virginia.edu

Samarth Swarup
BioComplexity Institute,
University of Virginia
Charlottesville, VA, USA
ss7rs@virginia.edu

Abstract

User behaviour Prediction from social media posts could potentially enable more robust active cognition models, and help construct more accurate simulations of the spread of online information to further the understanding of adversarial manipulation of such information. This study presents a network based framework to predict a given Twitter users reactions to a given set of information. We used natural language processing to tokenize each tweet from our collected data, and constructed a semantic network from given Twitter users timeline. Then, we used text tokens as nodes and calculated the unweighed centrality and weighted centrality with TF-IDF for tweets the user may see, and we used such values with other Twitter-specific features to train classifiers. The classifier takes a list of tweets and assign probabilities of different types of user behaviours. In our evaluation, the implementation of the semantic network has generally increased prediction accuracy from baseline models. We also provide several potential applications of our framework.

1 Introduction

1.1 Social Networks

The rapid emergence of various social media sites in the beginning of the 21st century has greatly transformed the lives of people worldwide by making them more connected than ever. Twitter is a microblogging and social networking service with 321 million monthly active users in 2018. On Twitter, a user may interacted with another user by like, retweet, reply, or quote the tweet posted by another user. Twitter provides researchers an excellent tool to study the many real-world implications of social interactions of contact networks and make predictions based on available data, such as predicting election outcomes[1], disease outbreaks[2], and stock returns[3]. However, Twitter research done by adversarial entities such as Russias Internet Research Agency recently has also raised public concerns of election interference and the spread of misinformation[4].

1.2 Semantic Networks

Semantic networks[5] are graphical representations of knowledge based on meaningful relationships of written text, structured as a network of words cognitively related to one another[]. In this study, nodes of the Semantic network are words that represent a variety of concepts found in each tweets posted by the user. The connections between nodes are referred to as edges which represent relationships between connected concepts. Semantic networks allow the extraction of meaningful ideas by identifying emergent clusters of concepts rather than analyzing frequencies of isolated words; in this way, analyzing online social media can enhance understanding of complex humans interactive behaviour. In constructing the semantic network, we also incorporated natural language processing (NLP)[6] techniques, to extract and tokenize information from each tweets using both human and computerized methods.

1.3 Spreading Activation

Spreading activation[7] is a method for searching semantic networks. The search process starts with the activation of a set of nodes and then iteratively propagating that activation out to other nodes linked to the source nodes. In this study, when nodes in the network are activated, e.g. the user interacted with another user by a quote, the activation spreads from the nodes that correspond to the words in the quote in the semantic network to nearby nodes. Spreading activation is also hypothesized as the model for the automatic activation of sentiments that may affect the interaction.

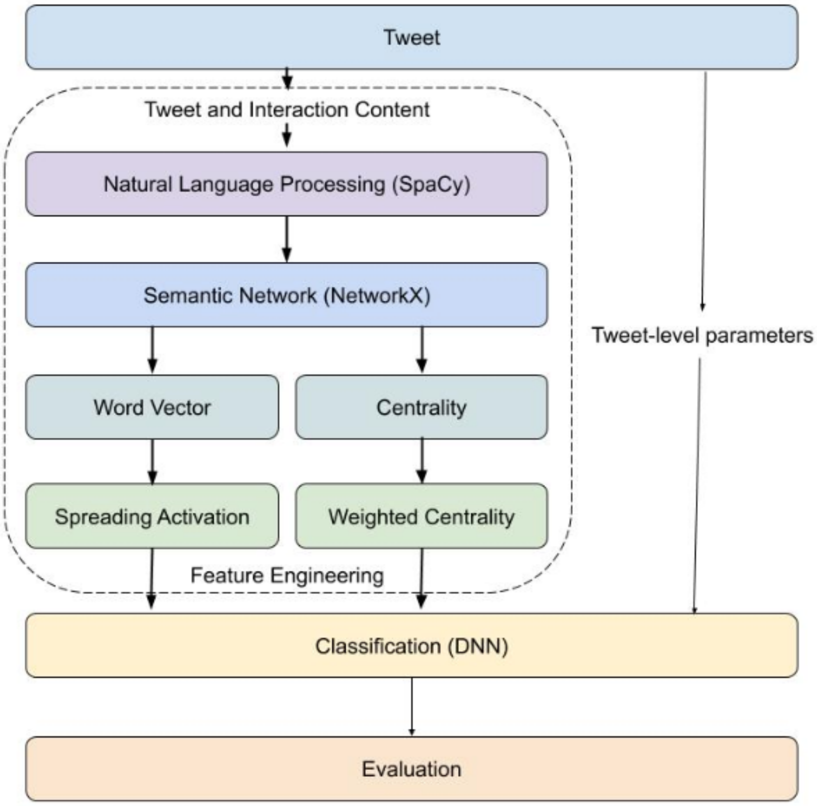

Figure 1: Overview

1.4 Word Embedding

Word embeddings[8] are a type of word representation that allows words with similar meaning to have a similar representation. They are a distributed representation for text that is perhaps one of the key breakthroughs for the impressive performance of deep learning methods on challenging natural language processing problems. In this study, we used word embeddings to vectorize individual words from tweets that are present in the semantic network, and weigh the word vector with the activation value from the spreading activation method.

1.5 Convolutional Neural Networks

Convolutional Neural Network(CNN)[9] is a class of deep neural networks that was inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. In this study, we implemented an input layer, an output layer, and multiple hidden layers with RELU activation function in order to produce a classifier that utilizes the semantic network approach to predict user interactions.

1.6 Objectives

This study seek to resolve the following premise:

> Given a Twitter user A, and a set of tweets $t_1, t_2, t_3 ... t_i$ the user A sees, can we predict how the user may react(reply, retweet, quote, like) to each tweet in the set?

We resolve the premise by collecting the tweets the user may see and constructing and analyzing semantic networks of the given users timeline data.

1.7 Public Significance

To combat distortions from foreign entities on Social media, DARPA has created the SocialSim challenge[10] in 2018 to develop innovative technologies for high-fidelity computational simulation of online social behavior, hoping it could enable a deeper and more quantitative understanding of adversaries use of the global information environment

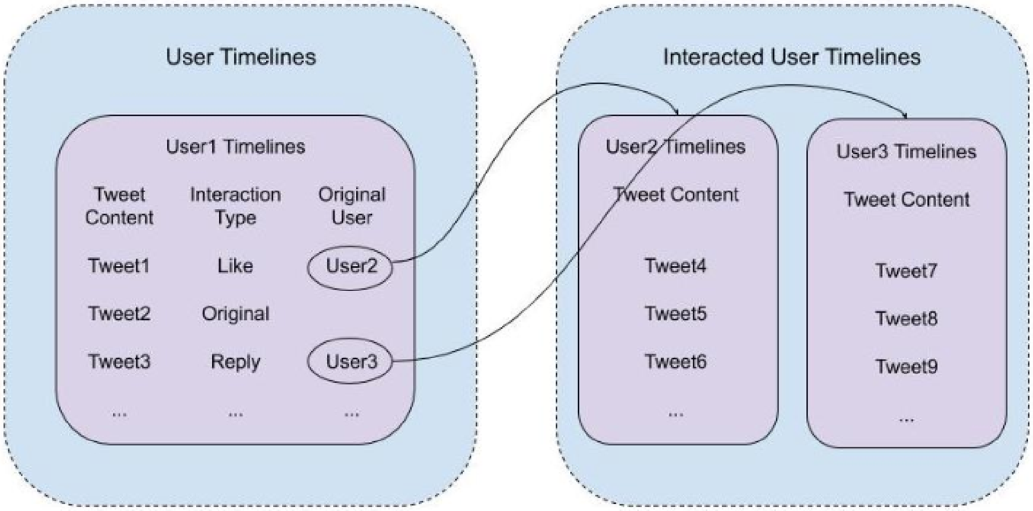

Figure 2: Data Collection

than is currently possible using existing approaches. This study propose a framework to further enhance the simulation of user behaviour by incorporating network analysis, and findings from this study could prove pivotal in informing and improving the simulation of social networks to help combat the manipulation of social networks by foreign entities.

2 Related Work

With the rising research interest with Twitter data, there are several work that slightly overlap with our work. With regard to prediction on Twitter, Petrovic, Osborne and Lavrenko[11] studied the problem of predicting whether a user will retweet a particular tweet, TwitterMancer[12] proposed predicting what type of interaction may take place between two different users with link prediction, and RealGraph[13] utilizes a framework to compute relationship strength for ties based on directed interactions between users. Our work differs with these prior work by being the first to construct a semantic network and utilize NLP techniques with Twitter data to predict how a given user may react to a set of tweets. For word embedding, Tang et al.[14] explored sentiment classification on Twitter with learning sentiment-specific word embedding. Our work differs with them in that we seek to weigh such embedded word vectors with activation values from the spreading activation method and evaluate the results. In addition, this work does employ a similar approach to Kang, Swraup[15] to utilize Semantic Network and Spreading Activation techniques for Twitter Analysis. However, this work seek to make predictions about user behaviours rather than analyze user sentiment on vaccines.

3 Methods

Our framework is implemented with Python. As shown in Figure 1, it has three major components: data collection, feature engineering, and classification. In section 3.1, we will detail the the methodology and the parameters we utilized to collect the necessary data. In section 3.2, we will detail the data preprocessing techniques, the generation of semantic network, the implementation of Spreading Activation with word embedding, and the computation of different centralitiy and weighted centrality using TF-IDF. In section 3.3, we will detail the steps we take to establish a baseline model for data classification and prediction, and the implementation of a CNN model for better classification accuracy results.

3.1 Data Collection

We utilized Twitter's open-source API to extract user data. The API allows us to retrieve up to 3200 of the most recent Tweets posted by a given user from user timeline, and up to 3200 of the most recent Tweets liked by a given user from users like timeline. However, if the user posted or liked more than 3200 tweets, the timespan for users timeline and like timeline may vary, results in missing data and undermines the validity of this study. Thus, we chose a set of criteria that guarantees the user we

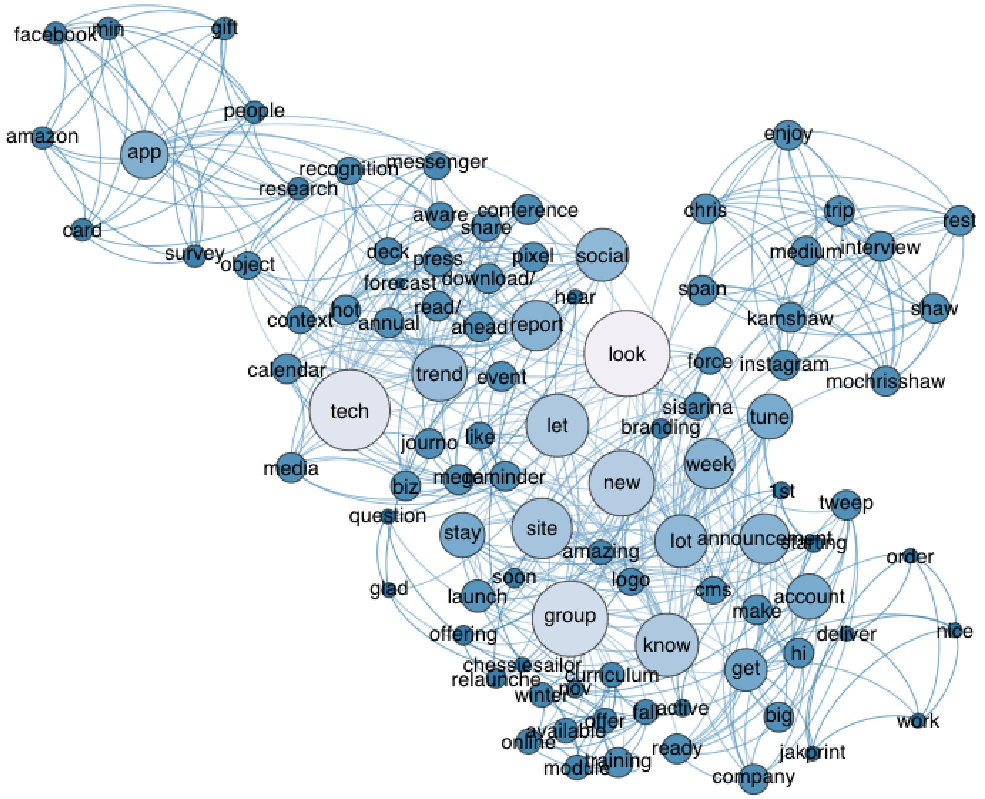

Figure 3: Sample Semantic Network(10 tweets)

selected with an appropriate number of tweets and likes in the users timeline and like timeline, respectively. We implemented a web crawler to download users timeline and like timeline with multiple parameters associated with each tweet. We merged users timeline and like timeline, and sorted the data with the created time of each tweet. We saved this data in document A. Then, we examined tweets that the user interacted with another user (like, reply, quote, retweet). We define such users who interacted with the given user as known interacted users. We downloaded all of such known interacted users timeline and like timeline, with the same sets of parameters associated with each tweet (Figure 2). Due to Twitters rate limit constraints, the timeline and like timeline for each known interacted user only consists up to 200 tweets each. The above data for all the known interacted users are stored in document B. We randomly sample an appropriate number of Twitter users that meet the criteria, and repeated the data collection process for every user.

3.2 Data Analysis

3.2.1 Data Preprocessing

For each tweet from document A or document B, we utilized SpaCy, an open-source Natural Language Processing software library to preprocess the tweet text. Our data preprocessing pipeline has three components. First, text tokenization is applied to segment the tweet into words, punctuation and so on. Then, words are transformed to their respective base forms to prevent the creation of multiple nodes of the same word, just in different forms, in the network. Finally, we filter out the punctuation and the stop words from the tweet to prevent useless data from interfering with the computation of centrality which may undermine the validity of the study. For example, the sentence

She just likes to drink tea.

is represented by a list of text tokens (She, like, drink, tea) after preprocessed by the pipeline.

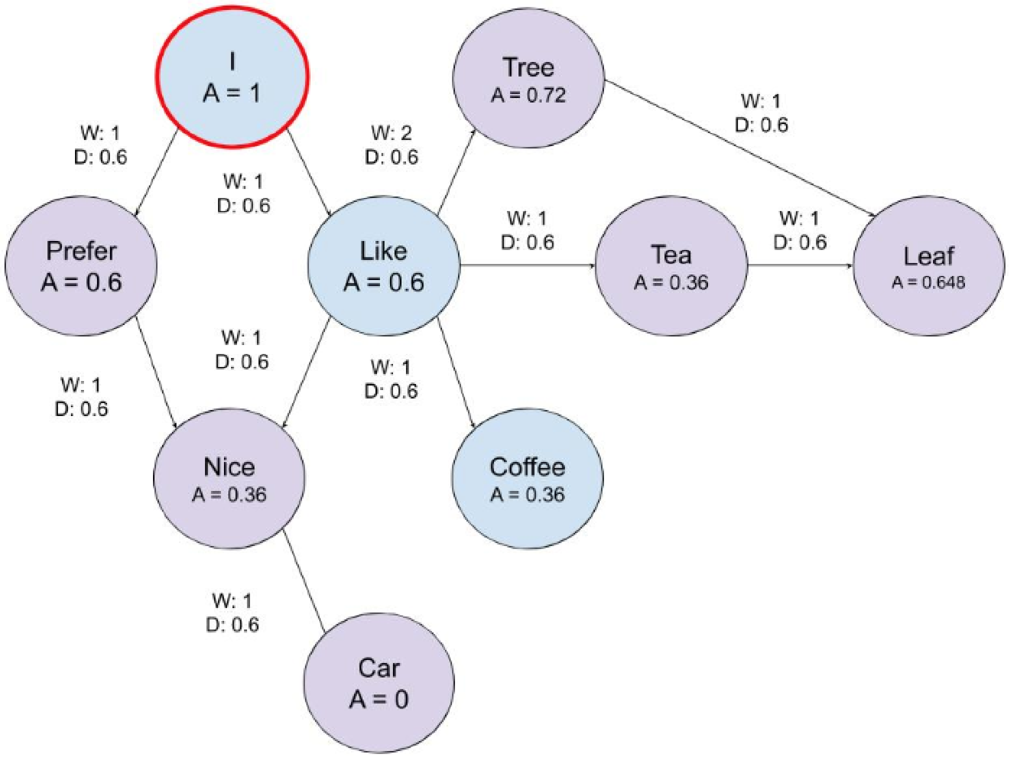

Figure 4: Spreading Activation with Semantic Networks

3.2.2 Semantic Network Construction

We used NetworkX to construct Semantic Networks. After defining the size and the scope of the network, new nodes (i.e., words tokenized from the prior step) are added to the network by adding tweets from document A to the network. When a node is added to the network, we would first examine if the network already contains nodes the represents the same word. If yes, this node is not added to the network, and we increase the size of the corresponding existing node already in the network by 1. Otherwise the node is added to the network. After all the nodes from the tweet are added or updated, we initialize edges to connect every node in the tweet. If an edge already exist between two nodes, the edge is not initialized, and we increase the weight of the corresponding existing edge by 1. We repeat this process for every tweet we intend to include in the network. Figure 2 shows a semantic network with 10 tweets.

3.2.3 Computation

For the target tweets we intended to analyze, we utilized concepts of Word Embedding and Network Centrality to compute the importance of individual words from target tweets in the Semantic Network.

Spreading Activation

$$\begin{aligned}
\mathrm{W}(v_i) &= \mathrm{W}(v_{i-1}) * \mathrm{W}(E(v_i, v_{i-1})) * \alpha \\
\text{where } \mathrm{W}(v_i) &= \text{Weight of node} \\
\mathrm{W}(v_{i-1}) &= \text{Weight of source node} \\
\mathrm{W}(E(v_i, v_{i-1})) &= \text{Edge Weight connecting node} \\
&= \text{and source node} \\
\alpha &= \text{decay value}
\end{aligned}$$

We implemented Spreading Activation with Semantic Network produced by NetworkX in prior the step to compute the weight of nodes (i.e., words) for the tweet we intended to analyze. We iterate through the text tokens of the tweet. If the network already contains nodes the represents the same word as the given text token, we activates the node by assigning weight of 1 to that node. Then, we iteratively propagate that activation out to other nodes linked to the source node in the Semantic Network, and assign these nodes with new weights.

The weights assigned to such new nodes are computed by the above formula.
An example of the spreading activation implementation is shown in Figure 4 (decay = 0.6, threshold = 0.4). Here, the tweet we intended to analyze is ”I like coffee”. First, the node ”I” is activated with an activation value of 1. It first propagate the activation value to nodes that are closest to ”I”. Thus, the nodes ”prefer” and ”like” are activated, both receiving a activation value calculated by the above formula, in this case 0.6. Then, these nodes propagate the activation value further in the semantic network, until all propagating nodes have a activation value that's lower than the threshold, in which the nodes stop propagating activation values to other nodes. At the time, the activation value is recorded for all nodes. Then, we activate the node ”like”, and repeat the above process for every word token in the tweet. Finally, we add all the activation values for each notde during each process. In addition, any node may receive activation values from more than one source nodes, and if the network does not contain nodes that represents the same word as the given text token, we do not activate any text tokens in the network.

Word Embedding After the activation value was calculated for each node in the semantic network, we proceed to calculate a weighted word vector for each node. We used spaCy's built-in word2vec function to vectorize the word token corresponding to each word vector, then we weighed each word vector using the formula below.

$$\begin{aligned}\text{weighted word vector} &= [v_1 * a, v_2 * a, v_3 * a]\\ \text{where } v_i &= \text{vector value}\\ a &= \text{activation value}\end{aligned}$$

The dimension for the word vector will be 300. Then, we calculated the average weighted word vector for each node in the network.

Network Centrality We used NetworkXs built-in API to compute multiple centrality concepts for the tweet we intended to analyze. We first iterate through the text tokens of the tweet. If the text token could be matched with any node in the Semantic Network(i.e. the tweet contains words that exist in the Semantic Network), we compute the centrality for that node in the Semantic Network. If no such match were found, the centrality for the text token is set to 0. We sum the centrality of each text token in the tweet to compute the unweighted centrality of the tweet. For weighted centrality, we also multiply each text tokens centrality with the TF-IDF values of the corresponding node to the text token. Here, the TF-IDF values of the node is computed by

$$\begin{aligned}TF - IDF &= TF(t, d) * IDF(t)\\ &= TF(t, d) * \log \frac{N}{df}\\ \text{where } t &= \text{text token t}\\ d &= \text{tweet d}\\ N &= \text{N tweets in the network}\\ df &= \text{df tweets containing corresponding}\\ &\quad\ \text{node of text token t}\end{aligned}$$

After calculating the TF-IDF values, the weighted centrality are calculated by

$$\begin{aligned}\text{weighted centrality} &= c_1 * t_1 + c_2 * t_2 + c_3 * t_3 ...\\ \text{where } c_i &= \text{centrality of text token i}\\ t_i &= \text{TF-IDF value of text token i}\end{aligned}$$

3.3 Data Prediction

3.3.1 Preparing Data

When a user interact with another user (like, reply, quote, retweet), we define the text that the user posted as user text, and we define the original tweet the user responded to by such interactions as original text. In document A, if the users tweet interacted with another user, we use the user text of previous m number of tweets this user posted prior to this interaction to establish a Semantic Network. Then, we use the original text of this tweet and compute the weighted and unweighted centrality. We store the centrality and other twitter parameters that associated with this tweet in our final dataset. We repeat this process for each tweet in document A that the user interacted with another user, and label each tweet with that type of interaction. In document B, we use the user text of tweets from this users known interacted users posts from the dataset, and use the user text of previous m number of tweets this user posted in document A prior to the created time of the post by that this users known interacted user to establish a Semantic Network. Then, we use the user text of this tweet and compute the weighted and unweighted centrality. We store the centrality and other twitter parameters that associated with this tweet in our final dataset. We repeat this process for k number of tweets in

Table 1: User Sample Overview

Parameters	Value
description	151.12
followers	1709.18
friends	1965.18
list	70.88
words per tweet	16.36
interaction	3571.88
original posts	1663.9
like	1727.54
reply	678.68
retweet	1079.7
quote	85.96

document B, and label each tweet as no interaction, meaning the user is likely to see the tweet but choose to not respond to it. We repeat the above process for every Twitter user that we collected its data and its known interacted users data.

3.3.2 Model Evaluation

We divide the final dataset to training and testing data. We first implemented a baseline model that only contains the tweet-level parameters in Table 1. Then, as shown in Table 2, we implemented several other models with different sets of parameters. The performance metric we selected was accuracy, or how accurate may the model predict user's interactions for a given tweet, and F-Score, which also considers the precision and the recall values to provide a more board explanation to the results.

$$\text{Accuracy} = \frac{\text{Correct Classification}}{\text{Total Population}}$$

We also computed the F-Score to for each type of tweet. The formula for F-Score is shown below.

$$\begin{aligned} \text{F-Score} &= \frac{2PR}{P+R} \\ \text{where } P = \text{Precision} &= \frac{TP}{TP+FP} \\ R = \text{Recall} &= \frac{TP}{TP+FN} \end{aligned}$$

All models are trained and examined with CNN.

4 Experiment

We analyzed the performance of our framework by performing real-world experiments. In section A, we will detail our setup and implementation of the experiment. In section B, we provide results of experiment and evaluate the improvements using our framework over baseline models.

4.1 Procedure

We randomly sampled 80 Twitter users that meet the specified criteria. Such users all have fewer than 3200 tweets in their timelines and like timelines. The criteria was chosen to avoid computing with incomplete data. As we limit the total number of likes and tweets from the user, we could guarantee that we collected the entire timeline and the like timeline of the user, without any missing data. Table I summarizes the general characteristic of the 80 Twitter users. Then, we downloaded their twitter timelines, like timelines, and merged two timelines. We sorted the combined timelines by the timestamp of each tweet. We then examined the combined timelines for all sampled users. In all, as shown in Figure 5, about 68.23 % of the sampled users' posts from their timelines were interactions, while about 31.77% of the sampled users' posts were original. For the tweets that are interactions, we labeled each tweet based on different types of interactions. Also, like was the most common interaction overall, in which about 48.37% of sampled users' interactions were likes, followed by retweet(30.23%), reply(19.00%), and quote(2.40%), respectively. Using these interactions, we downloaded the known interacted users timelines and like timelines using methods specified in Section 2.1. In all, we collected a total of 27 million tweets created before August 15, 2019. After the Twitter data were collected, we preprocessed the data by tokenizing all the tweets. Then, for every sampled user, we constructed the semantic networks for each interacted tweets and 2000 random tweets that the user may see but chose to not interact with. Although it was not possible for us to know what time did the user log in to Twitter,we used time intervals to estimate the likielihood that a tweet from a known interacted user might be seen by the user. The time interval was computed by

Table 2: Prediction Accuracy

Dataset	100	300
all centrality only	50.67%	51.40%
baseline only	86.62%	86.62%
baseline + subgraph centrality	86.39%	86.80%
baseline + weighted subgraph centrality	86.94%	87.07%
baseline + all centrality	85.27%	85.40%

$$\begin{aligned} \delta t &= t_{current} - t_{user} \\ \text{where } t_{current} &= \text{timestamp of the tweet} \\ t_{user} &= \text{timestamp of user's first tweet} \\ &= \text{after } t_{current} \end{aligned}$$

We incorporated the time interval feature into all models. Next, to examine if the size of the semantic network may enable the model to have better prediction accuracy, we constructed semantic networks with a size of 100 and 300 tweets. The size of the semantic network was chosen to reduce computation time, and also enable real-world applications on less powerful devices. First, word vectors were generated for each node in the network, and the spreading activation method was used to weigh the word vectors for each node, during the analysis of each target tweet. Then, we calculated the centrality values and the weighted centrality values with TF-IDF for each tweet, using the methods specified in Section 2. Table II summarizes the general characteristic of the tweets we analyzed. Finally, we divided the final dataset to training and testing data, and build CNN models to compare the prediction accuracy for each model.

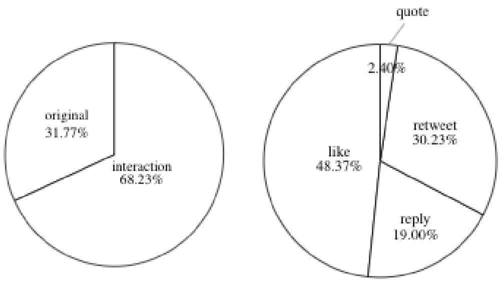

Figure 5: Sampled User's Activities on Twitter

4.2 Evaluation

The results of this evaluation are shown in Table III. On average, models with subgraph centrality achieved 0.5% increase prediction accuracy over the baseline model. When comparing the best model with an additional centrality and the baseline model, as shown in Table 2, the best model performs better than the baseline model for about 86.25% of the time when the size of the semantic network was 100, and the best model performs better than the baseline model for about 91.25% of the time when the size of the semantic network was 300. On average, the best model with an additional centrality achieved an average 0.857% increase prediction accuracy over the baseline model when the

Table 3: F-Score

	reply		like		retweet		quote	
Dataset	100	300	100	300	100	300	100	300
all centrality only	19.69%	21.63%	28.51%	28.92%	31.75%	31.99%	3.28%	3.92%
baseline only	56.47%	56.47%	73.91%	73.91%	62.79%	62.79%	10.09%	10.09%
baseline + subgraph centrality	55.05%	55.56%	71.47%	73.98%	61.38%	61.74%	12.54%	15.98%
baseline + weighted subgraph centrality	55.67%	55.87%	72.69%	74.85%	63.10%	63.25%	10.76%	11.97%
baseline + all centrality	52.58%	52.61%	68.66%	69.67%	59.52%	60.61%	8.63%	9.67%

size of the semantic network was 100, with a maximum improvement of 1.635%, and the best model with an additional centrality achieved an average 1.052% increase prediction

accuracy over the baseline model when the size of the semantic network was 300, with a maximum improvement of 3.917%. For F-Score, as shown in Table 3, models with both subgraph centrality and weighed subgraph centrality achieved about 1 2% improvement over the baseline model for retweet and quote.

5 Conclusion

In this paper, we presented a framework, TwitterPred, to predict user interactions on Twitter. We successfully incorporated semantic network analysis into our models, with additional concepts such as spreading activation, word embedding, network centrality, and TF-IDF. While being simple, our model has seen modest improvement in accuracy over baseline models, and the approach is easier to understand and requires less computational resources than traditional NLP methods.

References

[1] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan. 2012. A system for real-time twitter sentiment analysis of 2012 u.s. presidential election cycle. In Proceedings of the ACL 2012 System Demonstrations.

[2] Quincey, E., and Kostkova, P. 2010. Early warning and outbreak detection using social networking websites: The potential of twitter. In Electronic Healthcare. Springer Berlin Heidelberg.

[3] Bollen, J.; Mao, H.; and Zeng, X.-J. 2010. Twitter mood predicts the stock market. Journal of Computational Science 2(1):18.

[4] Stukal, D., Sanovich, S., Bonneau, R., Tucker, J. (2016). Detecting Bots on Russian Political Twitter. Manuscript submitted for publication

[5] J. F. Sowa, "Semantic networks", Encyclopedia Cogn. Sci., 2006.

[6] Daniel Jurafsky , James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall PTR, Upper Saddle River, NJ, 2000

[7] Collins, A. M., Loftus, E. F. (1975). A spreading-activation theory of semantic processing. Psychological review, 82(6), 407.

[8] Vilnis, Luke, and Andrew McCallum. "Word representations via gaussian embedding." arXiv preprint arXiv:1412.6623 (2014).

[9] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

[10] Blythe, James, et al. "The DARPA SocialSim Challenge: Massive Multi-Agent Simulations of the Github Ecosystem." Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[11] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Rt to win! Predicting Message Propagation in Twitter , 2011.

[12] K. Sotiropoulos, J. W. Byers, P. Pratikakis, and C. E. Tsourakakis. Twittermancer: Predicting interactions on twitter accurately. arXiv preprint arXiv:1904.11119, 2019.

[13] Kamath, K., Sharma, A., Wang, D. Yin, Z. (2014) RealGraph: User Interaction Prediction at Twitter. In User Engagement Optimization Workshop @ KDD

[14] Tang, Duyu, et al. "Learning sentiment-specific word embedding for twitter sentiment classification." Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2014.

[15] Kang, G. J., Ewing-Nelson, S. R., Mackey, L., Schlitt, J. T., Marathe, A., Abbas, K. M., Swarup, S. (2017). Semantic network analysis of vaccine sentiment in online social media. Vaccine, 35(29), 36213638. doi:10.1016/j.vaccine.2017.05.052