Analyzing User Discussion Dynamics in Social Media Platforms

Student Name: Arpan Mukherjee

IIIT-D-MTech-CS-DE-18-MT17007
June, 2018

Indraprastha Institute of Information Technology
New Delhi

Thesis Advisors
Dr. Tanmoy Chakraborty

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science,
in Data Engineering Category

Keywords: Community analysis, online discussion, user conversation dynamics, social media

Certificate

This is to certify that the thesis titled **"Analyzing User Discussion Dynamics in Social Media Platforms"** submitted by **Arpan Mukherjee** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by her under our guidance and supervision at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Tanmoy Chakraborty

Indraprastha Institute of Information Technology, New Delhi

Abstract

Understanding how users behave when they connect to social networking sites creates opportunities for better interface design, more productive studies of social interactions, and improved design of content distribution systems. Traditionally, user behavior characterization methods, based on individual features of users, are not appropriate for online networking sites. In these environments, users interact with the site and with other users through a series of multiple interfaces that let them upload and view content, choose friends, rank favorite content, subscribe to users and allow many other interactions. Different types of interactions can be observed among different types of users, and interactions on the same topic between a set of users can have different patterns as well. The motivation is to explore user behavior and the underlying conversation patterns. How a set of users react to a piece of particular news, or for a political leader how a single user changes the polarity with time or how it is constant throughout the time which can be observed in his expressive online statements (e.g., - tweets, Reddit discussion, Facebook comments, etc.). We study user behavior and its patterns in primarily three parts. (i) We present a novel quantification of conflict in an online discussion. Our measure of conflict dynamics is continuous-valued, which we validate with manually annotated ratings. Firstly, we predict the probable degree of conflict a news article will face from its audience. We employ multiple machine learning frameworks for this task using various features extracted from news articles. Secondly, given a pair of users and their interaction history, we predict if their future engagement will result in a conflict. (ii) We study how the sentiment of a user towards entities can be predicted using the tweets the user has posted so far. We consider all the available sentiments of the entities for the prediction task. (iii) In this task by using the same twitter data, we study how the sentiment of a user to an entity changes with respect to time - with time how a user changes the polarity, whether it remains same or it varies after a certain point, if so how many users had the same kind of polarity drift for an entity.

Acknowledgments

I would like to express my deepest gratitude to my advisor Dr. Tanmoy Chakraborty for his guidance and support. I would like to thank him for his mentorship at every stage of this thesis work. The door to his office was always open whenever I had a doubt. I would not have been able to complete my thesis work this smoothly in IIIT Delhi without his handholding and consistent support.

I would like to thank the support of my collaborators - Subhabrata Datta, Divam Gupta, Gunkirat Kaur, Shreyans Mongia and all the members of LCS2 lab.

I would like to thank my supportive family and friends who encouraged me and kept me motivated throughout the thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last decade, with the rapid rise of social media platforms, information sharing has been democratized. Social media platforms (e.g., Facebook[1], Twitter[2], Reddit[3]) have become an integral part of the virtual communities, which allow people from different parts of the world to interact and connect each other by creating profiles and sharing their views on the platform. Due to the intrinsic human nature, users are not expected to exhibit one single and straightforward behavior. Some are enthusiasts and express themselves by updating blogs daily and upload as many videos or photos as they can, whereas there are users that act like free-riders and want to enjoy the contents made publicly available.

With more and more people coming together in this virtual world, differences of opinions and conflict are an inevitability. Conflict may arise from several premises – partial knowledge, socio-political understandings, clash of cultural and moral positions, and many more. It can be raised and developed from purely virtual individual interactions as well as real-world happenings. Although any difference of opinion can be identified as a conflict, its actual aspects are versatile. It may manifest itself within a vast spectrum, from constructive debates with well-formed argumentation to degenerated, unhealthy cyber-bullying and abuse. Thus a better introspection into the complex dynamics of conflict over online discussion platforms may provide more useful insights to the data analytics and social computing community as well as help moderators of online platforms to identify and eliminate abusive conflicts and make the web a better place. Previous studies [32] on conflicts either treated it as a binary phenomenon; or identified controversy scores over topics and not between two text segments [20]. Sophisticated NLP tools may come handy in this content; however, one major downside is their lack of scalability in handling large-scale online data. In this work, we focus more on objective, argumentative conflict, rather than subjective, aggressive conflict.

Online discussion platforms, through the lenses of engagement conflict, become a more complex

[1] www.facebook.com
[2] wwww.twitter.com
[3] wwww.reddit.com

dynamical process when the system frequently interacts with external sources. In this work, the external source is online news. Reddit has a specific community, *r/news*, dedicated to discussing news articles from various online news sources. Users post their views regarding news report and are engaged in a discussion. Here a two-way conflict comes into play – users holding opposite opinions against a report and users containing contradictory views towards each other.

Not just Reddit, we also performed an experiment on explicitly collected data from Twitter during UK election and examined how users behave to specific entities or party, how do they react or what kind of sentiment they hold for a political party. We also examined how the discussion patterns of users who are very frequent about their expressive statements on social media platforms, how the sentiment of these users changes with time and what is the underlying structure of such user behavior is.

1.1 Outline

The primary focus of the thesis will be on conflict detection. The rest of the thesis is organized as follows: Section 2 discusses the conflict detection and the work we have done with our novelty. In Section 3, we discuss how we can detect the polarity of an entity by a user. In Section 4, we discuss the task of temporal sentiment prediction of a user given the previous behavior of that user for that specific entity. In Section 5, we conclude the thesis and mention about future work.

Chapter 2

Conflict Detection

2.1 Related Studies

Conflict in community interaction, which is the prime theme of this study, is a well studied problem in social network theory, psychology and sociology [33, 38, 40, 56]. Different models and valuable introspection have emerged from these studies, such as how people tend to adapt towards certain acquaintances after initial conflict, fission in small group of networks post conflict, emotional effects of conflict on individuals, etc. However, studies on its online counterparts are much recent. Most of the studies in controversy and polarization over social media are based on Twitter [12, 21]. Garimella et al. [20] proposed a graph-based approach to identify controversial topics on Twitter and measures to quantify controversy of a topic. They used 20 different hashtags to classify topics of conversation. Partitioning retweet, follow and reply graphs, they computed the controversy related to each topic. Their work suggested the inefficiency of content-based measurements of controversy, majorly attributed by short spans of texts in tweets and high noise. Guerra et al. [23] proposed a similar approach to measure polarization over social media; there data also is mostly based on Twitter. However, one must keep in mind that, the nature of conflict for microblogs is substantially different from that of discussion forums, primarily due to the size of the text. Kumar et al. [32] focused on Reddit to identify roles of conflict in community interactions [10]. They performed their study on 36,000 Reddit communities (subreddits), identifying relation between inter-community mobilization and conflict. Their study also includes patterns of how people 'gang up' on the verge of conflicting engagements. They predicted mobilizations between communities based on conflicts using user-level, community-level and text-level features. They achieved 0.67, 0.72 and 0.76 AUC using Random Forest, LSTM and ensemble of both, respectively. Our work can be thought of as another side of their story – while they focused on conflict as a inter-community phenomenon, we attempt to address its dynamics in a microscopic level, inside a single community.

Stance detection and opinion mining is closely related to conflict identification and measurement. Most of the previous works in stance detection are based on stance classification of rumors in Twitter [35,37,59]. Rosenthal and McKewon [47] proposed an agreement-disagreement

identification framework for discussions in Create Debate and Wikipedia Talkpages. They defined various lexical and semantic features from discussion comments and achieved an average accuracy of 77% on the Create Debate corpus. Zhang et al. [57] used discourse act classification on Reddit discussions to characterize agreement-disagreement over discussion threads. Dutta et al. [16] employed an attention-based hierarchical LSTM model for further improvement of discourse act classification on the same dataset.

News popularity prediction, though does not handle conflict explicitly, is related to this work as it deals with engagement dynamics of online news. Previous studies can be classified into two main heads of approach – popularity of news in social media platforms [43, 46, 54], and popularity of news on web in general [17, 29]. The second approach deals with the prediction problem unaware of inter-user network information, thereby excludes the explicit interaction of users with themselves and with the news sources. Popularity prediction models focus only on the degree of engagement a news gets, without concerning about the types of engagement, which is our focus in this work.

Link prediction on social networks, as we already stated, is closely related to our formulated problem of predicting future conflict between users. There is rich literature focusing on this task [1, 22, 34, 53]. Bliss et al. [6] used evolutionary algorithm for link prediction in dynamic networks. One important advancement in recent times for learning graph-based data is Graph Convolution Networks [14, 30]. Zhang and Chen [58] applied convolution on enclosing subgraphs for link prediction. Berg et al. [4] also defined recommendation as a link prediction problem and used graph auto-encoder using deep stacking of graph convolutional layers.

2.2 Dataset Collection

We crawled discussion threads containing at least one news link in the posts or comments from *r/news* subreddit, starting from 2016-09-01 to 2019-01-16. Out of 43,343 discussion threads crawled, we discarded threads containing less than 10 comments. The remaining 17,351 threads containing a total of 5,502,258 comments were used for the experiments. We also crawled news articles mentioned in the threads, resulting in a total of 41,430 news articles from 5,175 different news sources.[1]

To evaluate our conflict measurement strategy, we employed three expert annotators[2] to identify conflict between two given texts (articles/comments). We asked them to rate an interaction with higher conflict score than another if they found more elaborate opposition in the first one. We provided the annotators with multiple examples annotated by us. We asked them to annotate the conflict in $[0 - 10]$ scale such that non-conflicting and highly conflicting texts will receive 0 and 10, respectively. For any interaction where only negativity has been expressed (sarcasm, popular slang without mentioning to what or whom it is addressed), we asked the annotators to rate as 1. We compute final ratings as the average of the ratings received. A total of randomly

[1] The dataset containing the news articles is public.

[2] They were experts on social media and their age ranged between 25-40 years.

selected 3,734 news-comment pairs and 6,725 comment-comment pairs were annotated. The inter-annotator agreement based on Fleiss' κ [18] is 0.79.

2.3 Conflict Quantification

Given two text segments, we measure conflict between them as how much opposite sentiment they exhibit. Here, we use target-dependent sentiment measurement (TD-sentiment) as sentence-level sentiment may not be a good indicator of stance towards a motion. Let us take the following two sentences:

1. *Applauds for the writer to rightly explain why immigration is not a real problem.*
2. *This is an extremely good analysis of why immigration should be stopped.*

Both of these sentences have positive sentence-level sentiment, though they carry conflicting opinion towards immigration. TD-sentiment for the term '*immigration*' is neutral for sentence 1 and negative for sentence 2. From this, we can conclude that these two sentences are potential indicator of conflict.

As defined in our problem statement, we compute conflict between news article and platform users as well as between pair of users. Firstly, we compute a set of keywords from our dataset (comments + news articles). We tag the sentences using Spacy[3] parts-of-speech tagger and collect nouns only, after removing stopwords and lemmatization. To handle co-references of persons, we substitute nominal pronouns '*he*' and '*she*' by the last named-entity found with 'Person'-tag. We include all the named entities in our keyword set, and top 60% of the rest, ranked in order of tf-idf values. This results in a final corpus-wide term set T.

Next, we compute TD-sentiment of news articles and comments using Multi-Task Target Dependent Sentiment Classifier (MTTDSC), a state-of-the-art deep learning framework proposed by Gupta et al. [25, 26] recently. MTTDSC is informed by feature representation learnedd for the related auxiliary task of passage-level sentiment classification. For the auxiliary task and main task, it uses separated gated recurrent unit (GRU), and sends the respective states to the fully connected layer, trained for the respective task. The model is trained and evaluated using multiple manually annotated datasets [15, 48, 52].

Let a document D (a single comment or a news article) be a sequence of sentences $[s_1, s_2, \cdots, s_n]$ and $T_D \subset T$ be the keyword set present in D (where T is the corpus-wide term set defined earlier). For any $t \in T$ occurring in s_i, MTTDSC computes a three class probability (positive, negative and neutral) vector v_t^i. Then for all the occurrences of t in D, we compute aggregate sentiment of D towards t as $S_{D,t} = argmax(\frac{1}{n}\sum_i v_t^i)$, $S_{D,t} \in \{1, 2, 3\}$, where negative, neutral and positive sentiments are represented by 1, 2 and 3 respectively. Following this, we construct a vector TS_D

[3] https://spacy.io/usage/linguistic-features

of size $|T|$ such that,

$$TS_D[i] = \begin{cases} S_{D,T[i]} & \text{if } T[i] \in T_D \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

TS_D now represents the aggregate sentiments of document D towards all the terms present in it. For any two documents D_1 and D_2, we then compute the *conflict factor* (cf) between them using their aggregate TD-sentiment vectors TS_{D_1} and TS_{D_2} as follows:

$$cf(D_1, D_2) = \sum_{i=0}^{|T|} min(TS_{D_1}[i], TS_{D_2}[i], 1)|TS_{D_1}[i] - TS_{D_2}[i]| \tag{2.2}$$

The component $min(TS_{D_1}[i], TS_{D_2}[i], 1)$ returns 0 when either of the i^{th} terms of TS_{D_1} and TS_{D_2} are 0, i.e., the term is not common, and 1 otherwise. This excludes terms which are not present in either of the texts to contribute to conflict computation. The value of the component $|TS_{D_1}[i] - TS_{D_2}[i]|$ can be 0 (when both texts have same sentiment towards the term), 1 (when one of texts hold neutral sentiment and other one positive or negative) and 2 (when texts hold opposite sentiments).

2.4 News-User Conflict Prediction

Given a news article N and the set of all comments C related to N, we define *News Conflict Score* as,

$$nc(N) = \frac{1}{|C|} \sum_{c \in C} cf(N, c) \tag{2.3}$$

This is a normalized score referring to what degree users oppose the views presented in the news article. We then extract following features from news texts to predict this score given a news article:

1. **TD-sentiment vector**, entity-wise sentiment expressed in the news, as we compute TS_D in Eq. 2.1.

2. **Count of positive, negative and neutral words**, tagged using SenticNet [9].

3. **Cumulative entropy of terms**, given by,

$$p = \frac{1}{|T|} \sum_{t \in T} tf_t(\log |T| - \log(tf_t))$$

   where T is the set of all unique tokens in the corpus, and tf_t is the frequency of term t in the news text.

4. **Fraction of controversy and bias words**, measured using the lexicon sets General Inquirer[4] and Biased Language[5]; we use the fractions of these lexicons present in the article as controversy and bias features.

5. **Latent semantic features** using ConceptNet Numberbatch pretrained word vectors[6] [51]; we compute TF-IDF weighted average of the vectors of the words present in an article to represent latent semantics of the article.

6. **LIX readability** [5], computed as: $r = \frac{|w|}{|s|} + 100 \times \frac{|cw|}{|w|}$, where w and s are the sets of words and sentences respectively, and cw is the set of words with more than six characters. Higher value of r indicates harness of the users to read the article.

7. **Gunning Fog** [24], computed by: $0.4 \times (ASL + PCW)$, where ASL is the average sentence length, and PCW is the percentage of complex words. Higher value of this index indicates harness of the users to read the article.

8. **Subjectivity**, calculated using TextBlob[7]. Its values lie in the range [0,1].

To predict the conflict score $nc(N)$, we use three regression models: **Lasso**, **Random Forest Regressor**, and **Support Vector Regressor**.

2.5 Inter-user Conflict Prediction

As already stated, we define the inter-user conflict prediction as a binary classification task to decide whether two users will engage in a conflict given their previous engagement history. We represent engagement history as a weighted undirected graph $G = \{V, E, W\}$, where every node $v_i \in V$ represents a user u_i, and every edge $e_{ij} \in E$ connects two nodes v_i, v_j if and only if u_i and u_j have been engaged with each other earlier (i.e., either of them have commented in reply to at least one comment/post put by other). Every edge e_{ij} is accompanied by a weight $w_{ij} \in W$ equal to the average conflict between u_i, and u_j, which is computed as follows:

$$w_{ij} = \frac{1}{N_{ij}} \sum_{k=0}^{N} cf(D_k^i, D_k^j) \tag{2.4}$$

where D_k^i and D_k^j represent the comments posted by u_i and u_j, respectively at their k^{th} interaction, and N_{ij} is the total number of such interactions already occurred. $cf(D_k^i, D_k^j)$ is computed following Eq. 2.2.

To predict conflict between user pairs, we propose four different frameworks: one using graph convolution and three using Support Vector Machine (SVM) with different feature combinations.

[4] http://www.wjh.harvard.edu/~inquirer/homecat.htm
[5] http://www.cs.cornell.edu/~cristian/Biased_language.html
[6] https://github.com/commonsense/conceptnet-numberbatch
[7] https://textblob.readthedocs.io/en/dev/

2.5.1 Graph convolution on engagement network

As typical user-user engagement networks of online discussion platforms are huge in size, we need to implement graph convolution over a subgraph. To predict the engagement type between a pair of users corresponding to vertices v_i and v_j, we compute an enclosed subgraph $G_{sub} = \{V_{sub}, E_{sub}\}$ containing v_i, v_j from G such that $\forall v_k \in V_{sub}$, $dis(v_i, v_k), dis(v_j, v_k) \leq dis_{max}$, where $dis(v_i, v_k)$ is the length of the shortest path between v_i and v_k, and dis_{max} is a threshold distance (see Section 2.6 for more details). All the edges in E_{sub} share the same weight as in G.

We compute the adjacency matrix $\mathbf{A}$ from G_{sub} as follows:

$$\mathbf{A}[i][j] = \mathbf{A}[j][i] = \begin{cases} w_{ij} & \text{if } e_{ij} \in E_{sub} \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

We represent every node v_i with a d-dimensional feature vector $x_i \in \mathbb{R}^d$, which represents previous commenting history of user u_i. We compute x_i as the average over all the feature vectors corresponding to previous comments from u_i, using the same feature selection method as in Section 2.4 with an additional feature as follows – a binary vector representing the news sources the user is engaged with. This leaves us with a tensor representation of user vertex features $\mathbf{X} = \{x_1, x_2, \ldots, x_{|V|}\}$.

The adjacency matrix $\mathbf{A}$ and the vertex feature tensor $\mathbf{X}$ now represent network history and comment history of all the users at an instance, respectively. First, we learn a lower dimensional feature representation $\mathbf{X'}$ from $\mathbf{X}$ as follows:

$$\mathbf{X'} = \sigma_r(\mathbf{K_f}^\top \mathbf{X} + \mathbf{B_f}) \tag{2.6}$$

where $\mathbf{K_f}$ and $\mathbf{B_f}$ are kernel and bias matrices to learn while training and $\sigma_r(x) = max(x, 0)$. We fuse these two histories together using graph convolution. We compute a degree-normalized adjacency matrix $\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{D}$ is the degree matrix of $\mathbf{A}$. This multiplication normalizes the effect of neighboring vertices so that higher degree vertices do not get over-weighted. Now, our convolution at the m^{th} depth is computed as,

$$\mathbf{H_{m+1}} = \sigma_r(\hat{\mathbf{A}} \cdot \mathbf{H_m} \cdot \mathbf{K_g^m}) \tag{2.7}$$

where $\mathbf{K_g}$ is the graph convolution kernel to be learned while training, and $\mathbf{H_m}$ and $\mathbf{H_{m+1}}$ are the input and the output for the m^{th} convolution respectively. Since we use three consecutive convolution layers, the final feature representation is $\mathbf{H_3}$.

For predicting whether there will be a conflicting engagement between users u_i, u_j, we select the i^{th} and the j^{th} feature vectors of $\mathbf{H_3}$ and compute a score $y \in (0, 1)$ as follows:

$$\mathbf{E} = [\mathbf{H_3}[i], \mathbf{H_3}[i]] \tag{2.8}$$

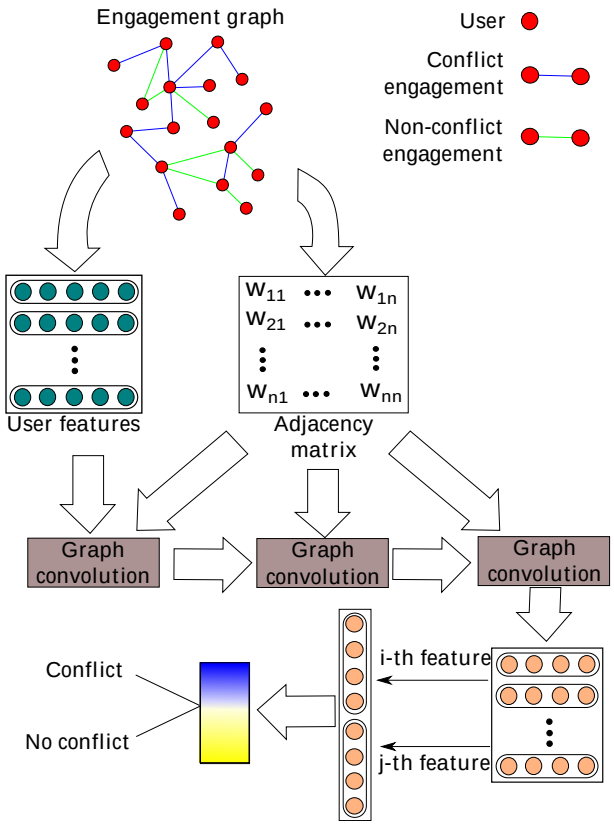

Figure 2.1: Inter-user conflict prediction using graph convolution.

$$y = \sigma_s(\mathbf{K_c}^\top \cdot \mathbf{E} + \mathbf{B_c}) \tag{2.9}$$

where $[\cdot, \cdot]$ stands for the concatenation operator, $\mathbf{K_c}$ and $\mathbf{B_c}$ are the kernel and bias for the classification layer respectively, and $\sigma_s(x) = (1 + e^{-x})^{-1}$. The complete architecture of the model is illustrated in Figure 2.1. This model is trained to minimize cross-entropy loss between true and predicted labels.

2.5.2 SVM-based frameworks

Graph convolution automatically learns feature representation for the interaction between user pairs from node features and connectivity of the nodes. For SVM, we need to manually identify interaction features. We extract the following textual and network based features for each user pair u_i, u_j:

1. **Count of relevant common tokens** from the previous comments of the users; we take the sum of tf-idf values of common unigram and bigrams in the comment history of both the users.

2. **Conflict vector** CV_{ij} between the pair computed using TD-sentiment vector TS_D following Eq. 2.1; given previous N_i^k comments of user u_i, $\{C_0, C_1, \ldots, C_{N_i^k}\}$ where the term $T[k]$ appear, we compute TS_{u_i}, the target sentiment vector of u_i averaged over the history

as,

$$TS_{u_i}[k] = \frac{1}{N_i^k} \sum_{l=0}^{N_i^k} TS_{C_l}[k] \tag{2.10}$$

We compute CV_{ij} as the element-wise absolute difference between TS_{u_i} and TS_{u_j}.

3. **Common news sources**, CN_{ij} taken as a vector of length equal to the number of news sources; for news source k, $CN_{ij}[k]$ indicates the number of articles from this news source where u_i, u_j both are engaged.

4. **Common discussions**, indicating the count of discussions where both u_i and u_j are engaged.

5. **Previous mutual engagement**, the total number of previous interactions between u_i and u_j.

6. **Previous conflict**, the average of mutual conflicts between u_i and u_j for their previous engagements.

7. **Neighbor interactions**, the count of conflicting and non-conflicting engagements for each user with its neighbor nodes.

We use three SVMs with Gaussian kernel – first SVM uses all the features mentioned above (SVM-all), the second one (SVM-text) uses only text based features (features 1 and 2) and the third one (SVM-net) uses only network based features (features 3-5). SVM-net, which has been used for negative link prediction by Wang et al. [53], serves as our external baseline.

2.6 Experimental Results

For the news-user conflict prediction task, total size of our feature vector is $8,136$. On a total set of $41,430$ news articles, we used $80:20$ train-test split keeping the fractions of different news sources same over train and test data.[8]

For the user-user conflict prediction task, the number of features representing user nodes in the graph convolution model is $8,236$. To construct enclosing subgraphs from user-user engagement network, we set the value of d_{max} (defined in Section 2.5.1) to be 100. This results in adjacency matrices with an upper bound of 5000 nodes. [9] We perform this prediction on 25 instances of the dynamic user engagement network, taking a total of $1,637$ different subgraphs from these instances. For any user pair on these subgraphs, if there is a conflicting engagement between them over an interval of next 24 hours, we label them as positive, otherwise negative. We take $213,998$ different user pairs altogether, randomly sampling equal numbers of positive and negative labels

[8] We used scikit-learn framework (`https://scikit-learn.org/stable/`) to implement all the regression models mentioned.

[9] We implement this model using Keras (`https://keras.io/`) and Tensorflow frameworks (`https://www.tensorflow.org/`).

Conflict type	RMSE	MAP	MRR
News-comment conflict	0.96	0.77	0.86
Comment-comment conflict	0.79	0.83	0.91

Table 2.1: Evaluation of conflict measurement on manually annotated conflict ratings.

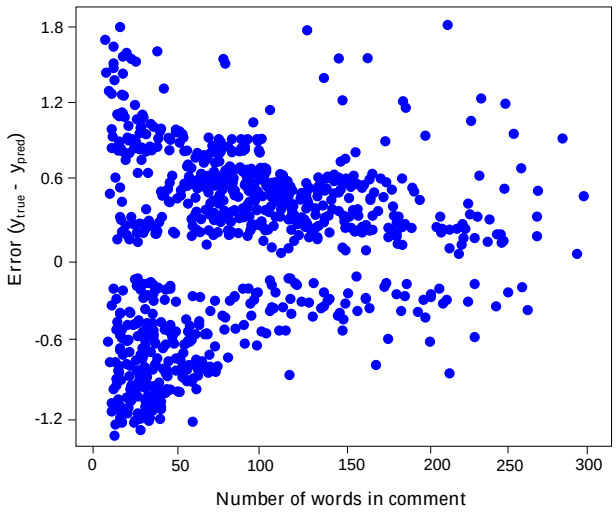

Figure 2.2: Error in conflict score vs. size of comments in words.

to avoid bias. Here again, we split the samples into 80 : 20 train-test splits, with 15% of the train data used as the development set to tune the parameters. We use Nadam (Adam with Nesterov momentum) optimization to train the model, with a batch size of 256.

2.6.1 Evaluation of conflict quantification

We test our conflict measurement on the manually annotated news-comment and comment-comment pairs (Section 2.2). To deal with different ranges, we normalize the cf values to the $[0 - 10]$ interval and measure Root Mean Squared Error (MSE). We also consider ranking comments accordingly to their conflicting tendency towards a particular news article and a particular comment. We compute the Mean Average Precision (MAP) of the ranking and Mean Reciprocal Rank (MRR) for top ranking position based on the ground-truth annotation mentioned in Section 2.2.

As observed in Table 2.1, measuring inter-comment conflict is rather an easier task compared to news-comment conflict. The feedback obtained from the annotators reveal that as most news articles are written in an objective style with less explicit opinion, it is hard to apprehend whether a comment holds opposite opinion to the news.

As there is no previous work in quantifying conflict between two text documents over online discussions, we implement the agreement-disagreement detection models proposed by Rosenthal and McKewon [47] (**Baseline-I**) and Dutta et al. [16] (**Baseline-II**). Baseline-I performs a three-class classification: *agreement*, *disagreement* and *none*. We identify disagreement as conflict and rest of the classes as non-conflict. We also define the probability of the disagreement class (predicted by Baseline-I) for an interaction as a unit norm score of conflict. Similarly, Baseline-II

Evaluation	SVM-all	SVM-text	SVM-net	GCN
Acc.	**0.89**	0.64	0.85	0.87
AUC	**0.89**	0.62	0.84	0.86
Acc. (new)	0.67	0.43	0.67	**0.72**
AUC (new)	0.65	0.43	0.65	**0.69**

Table 2.2: Evaluation of all the models for user-user conflict prediction. Accuracy is abbreviated as Acc. Acc. (new) and AUC (new) signify evaluation results for user pairs with no previous interactions.

performs a ten-class classification of discourse acts, from which we identify the classes *disagreement* and *negative reaction* together as conflict, and rest of the classes as non-conflict. Sum of the probabilities of these two mentioned classes is defined as unit norm conflict score predicted by Baseline-II.

We compare our strategy of conflict score prediction with the baselines through a three-way evaluation strategy:

1. We define a binary classification of interactions into conflict and non-conflict, evaluated using ROC-AUC;
2. We define a regression of the degree of conflict, where we scale the outputs of each model to the interval $[0, 10]$ and evaluate using RMSE;
3. We define a ranking problem of the interactions according to their degree of conflict, and evaluate using MAP.

As both the baselines perform their corresponding tasks (stance classification and discourse act classification) on discussion data, we perform this comparisons only for the comment-comment conflict prediction.

Table 2.3 shows that our proposed strategy outperforms both the baselines for ranking and regression tasks. This is quite expected as both the baseline models are actually classification frameworks. For the binary classification of conflicting and non-conflicting interactions, our strategy ties with Baseline-I.

Figure 2.2 plots the variance of error in conflict score with the change in the comment length. For news-comment pairs, we only take the comment length, while for comment-comment pairs we take the average of the length of both comments. To see whether the error in our score has any bias towards underestimation / overestimation, we take the difference $(y_{true} - y_{pred})$, where y_{true} and y_{pred} are manually annotated score and computed cf respectively. As we can see in Figure 2.2, our computed score underestimates conflict when comments are short, and overestimates as the size grows (more negative errors for size approximately less than 60 words; more positive errors afterwards). Also, absolute error rate decreases with increasing size of comments.

Such error pattern can be explained from the definition of conflict measurement itself. We use the sum of the absolute differences of sentiment towards specific targets common in documents,

Metric	Our method (conflict factor)	Baseline 1	Baseline 2
AUC	0.79	0.79	0.62
MAP	0.83	0.61	0.55
RMSE	0.79	1.67	2.09

Table 2.3: Comparison of conflict score with baselines.

Model	MSE	RMSE	sMAPE
Random Forest	6.194	2.489	0.099
SVR	4.041	2.010	**0.077**
Lasso	**3.179**	**1.783**	0.080

Table 2.4: Performance of different regression algorithms for news-user conflict prediction.

as conflict score which increases with the number of common targets present. As the length of the comments increases, the common word set also increases, and small differences add up to large conflict scores. For short comments, the number of common targets are also small, and the score tends to reflect less conflict than actual. For shorter comments, another problem is the use of semantically similar words occurring as targets in any of the comments in a given pair. For example, the sentences '*We do not support Democrats*' and '*We support Hilary*' are actually conflicting, as the targets *Hilary* and *Democrats* are semantically similar. But due to no common words, these pairs will be identified as non-conflicting.

However as our dataset suggests, the fraction of comments having greater than 50 words is 0.79; and the ratio between the number of words and targets is 17.678. This is particular to the online discussion forums, where users tend to get engaged in an elaborate manner, and therefore reduces the error margin of our conflict score. Our model achieves 0.96 and 0.79 RMSE for news-comment and comment-comment conflict, respectively, over the interval $[0, 10]$ which might be considered as significantly accurate for conflict modeling.

2.6.2 Evaluation of news-user conflict prediction

In our dataset, the news conflict scores (computed using Eq. 2.3) of the news articles vary from 0 to 138.15. In Table 2.4, we present the MSE (Mean Squared Error), RMSE and sMAPE (Symmetric Mean Absolute Percentage Error) for predicting news conflict scores using different regression algorithms. In terms of MSE and RMSE, Lasso regression performs the best, while SVR is the best performing one when evaluated using sMAPE.

We check to see which features are given more importance by our best performing regression algorithms. As we can see in Figure 2.3, term dependent sentiments are the most useful ones to predict how much likely is a news article to get negative feedback. In fact, this feature achieves way more importance compared to its next competitor, which again are polarity-oriented features. Interestingly, the count of negative polarity words has higher importance than the count of positive polarity words. The high importance of polarity related features may signify that news report expressing polarized bias tends to get more conflicting remarks. Readability indices (Gunning-Fog and LIX), albeit low, play some role in the prediction task. In fact, Gunning-Fog

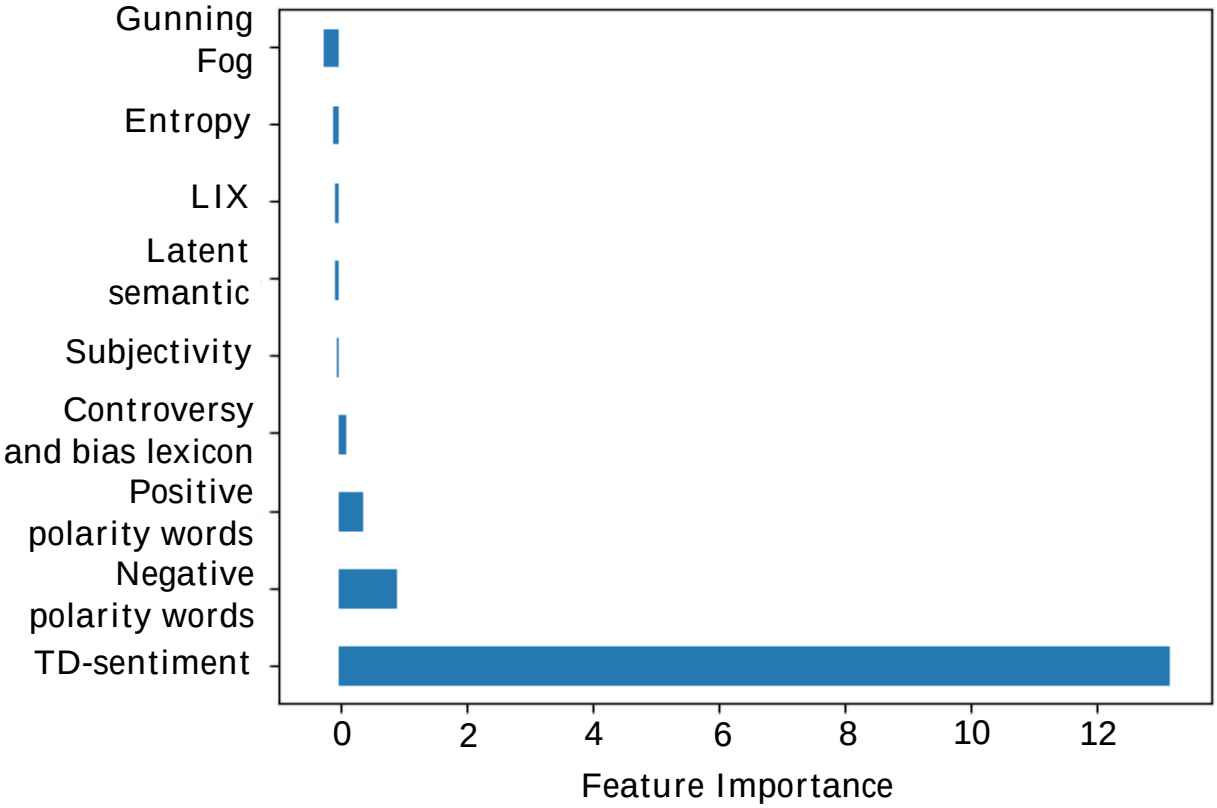

Figure 2.3: Importance of different features for news-user conflict prediction.

is substantially more useful compared to LIX.

2.6.3 Evaluation of inter-user conflict prediction

We evaluate all four models for two cases: (i) whole of the test data where a pair of users may or may not have previous interaction history, and (ii) user pairs who have no interaction history before the prediction instance. We present the evaluation results in Table 2.2. For the whole test data, SVM model with all the features performs the best. It is readily conclusive that, network-based features are of greater importance compared to text-based features for this task.

However, when there is no previous interaction history between two users, graph convolution beats all the models by a substantial margin. In fact, when there is no previous engagement history between users, the only feature available to the SVM model is the neighbour interactions; which means SVM-all and SVM-net actually become the same model, and SVM-text becomes a model with all zero features with all zero output.

2.7 Conflict Dynamics

We introspect into the dynamics of conflict in *r/news* community using the conflict measurements that we propose in Eq. 2.2 (for inter-user conflict) and Eq. 2.3 (for an aggregate conflict that a news article receives from the users).

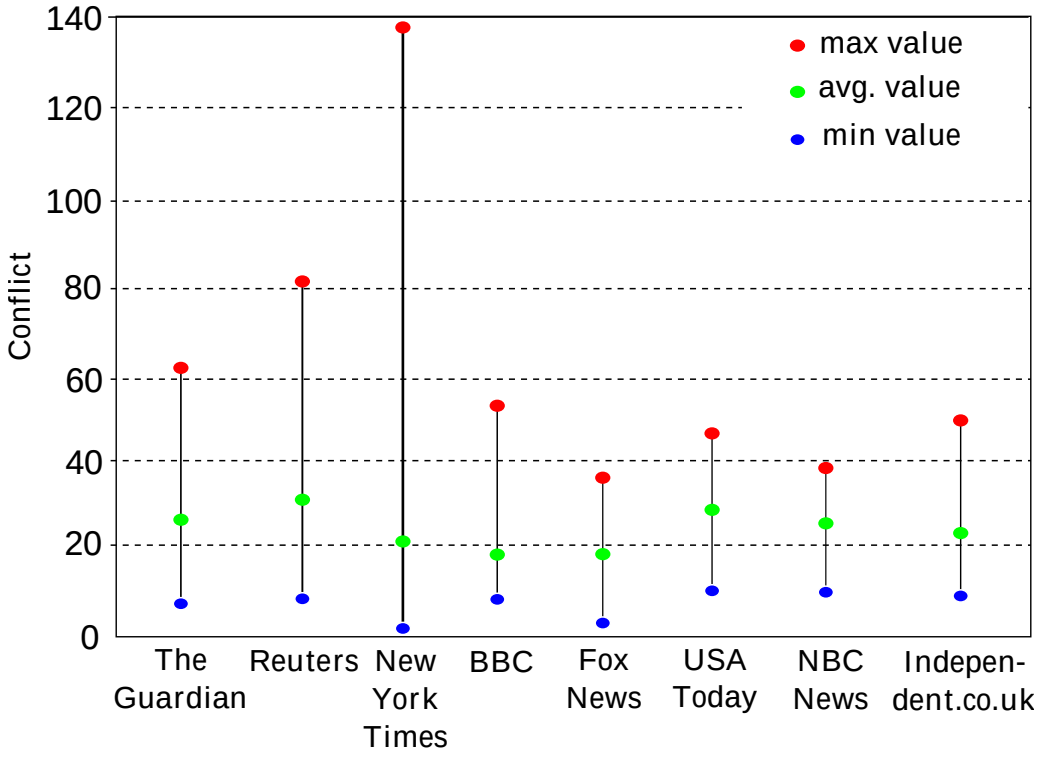

Figure 2.4: Distribution of maximum, minimum and average conflict scores for different news sources. This plot is for only top 7 news sources (ranked by number of articles).

2.7.1 Patterns of conflict for different news sources

Different news sources tend to face different degree of conflict from the users. In Figure 2.4, we plot maximum, minimum and average news conflict for different news sources in our dataset. Although the average conflict for different sources is in a comparable range, maximum values vary greatly. News sources such as *Fox News*, *USA Today* or *NBC News* maintain a sustained negative response, whereas *New York Times* or *Reuters* provoke sharp outrage at the some point.

We find that this outrage is signified by an article published in NYTimes on Dec 1, 2017, titled *Michael Flynn Pleads Guilty to Lying to the F.B.I. and Will Cooperate With Russia Inquiry*[10]. Figure 2.5 also indicates the sharp peak for New York Times corresponding to this article. The Guardian, Fox news and NBC News have similar peaks (red-circled) at nearby time instance, all corresponding to articles related to the same event. One can draw an intuitive correlation between the posting time of the article in the forum and the rise in conflict. It is important to note that at the time of posting, we identify the time when the news appeared on Reddit, not the time of its appearance on web.

[10] https://www.nytimes.com/2017/12/01/us/politics/michael-flynn-guilty-russia-investigation.html

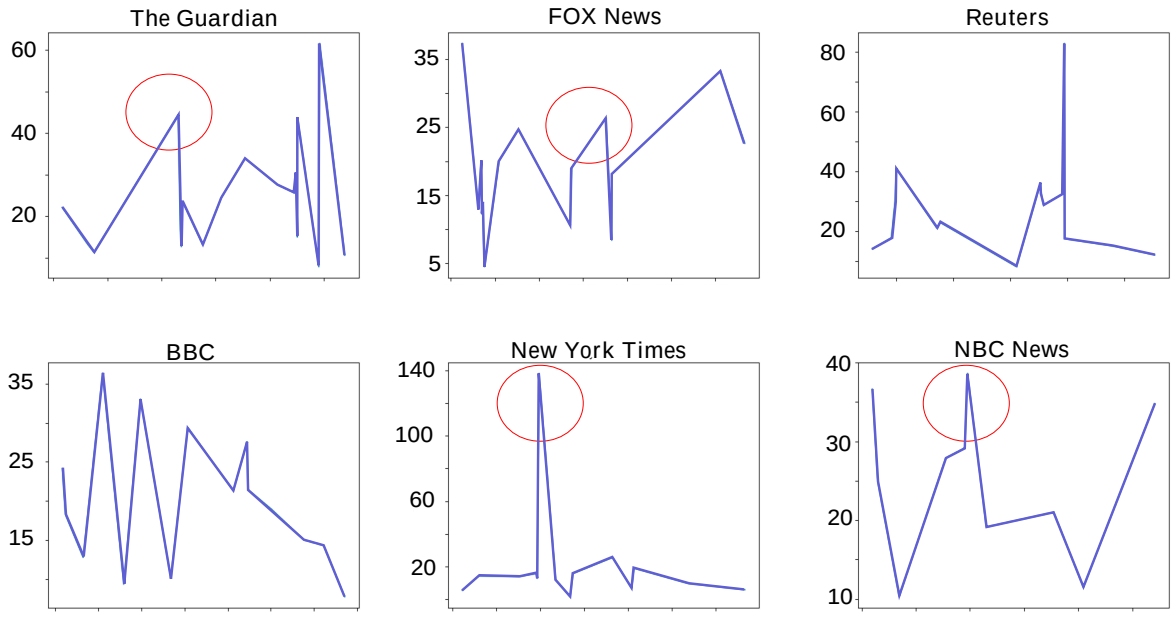

Figure 2.5: Temporal variation of news-user conflict for various news sources; conflict score and time are represented in y- and x-axis respectively. All the plots have time frame starting from Nov 17 - Dec 28, 2017.

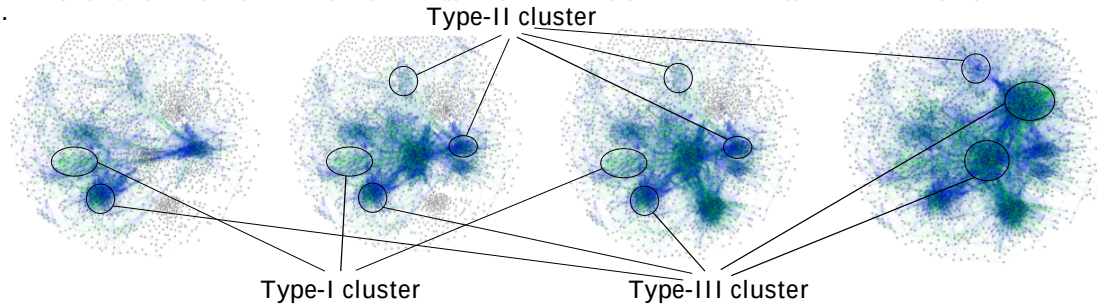

Figure 2.6: Snapshots of cluster formation in user-user engagement graph (left to right); blue and green edges correspond to controversial and non-controversial engagements respectively.

2.7.2 Engagement dynamics and inter-user conflict

To explore how conflict effects user engagement over *r/news*, we construct a temporal graph $G'(t) = \{V'(t), E'(t)\}$, where $v_i(t_i) \in V'(t)$ corresponds to user u_i who is engaged in a discussion at time t_i for the first time. For every pair of users (u_i, u_j) engaging with each other (anyone of them commenting in reply to the other) at time t_{ij}, there is an edge $e_{ij}(t_{ij}) \in E'(t)$. For better visualization, we classify edges as conflicting (blue) and non-conflicting (green), and plot only a subgraph using 5000 vertices. We use Fruchterman Reingold layout algorithm [19] on Gephi [2] to plot the graph and DyCoNet [28] to identify communities. In Figure 2.6, we present snapshots of the evolving graph. Each snapshot is taken at a time difference of approximately 24 hours, presenting a 4-day long abstraction through this engagement subgraph.

We can observe the formation of separate user clusters in terms of engagement. It is interesting to see that there are some clusters where users are predominantly engaged with each other in a conflicting manner (blue regions) and some in a non-conflicting manner (green regions). We also

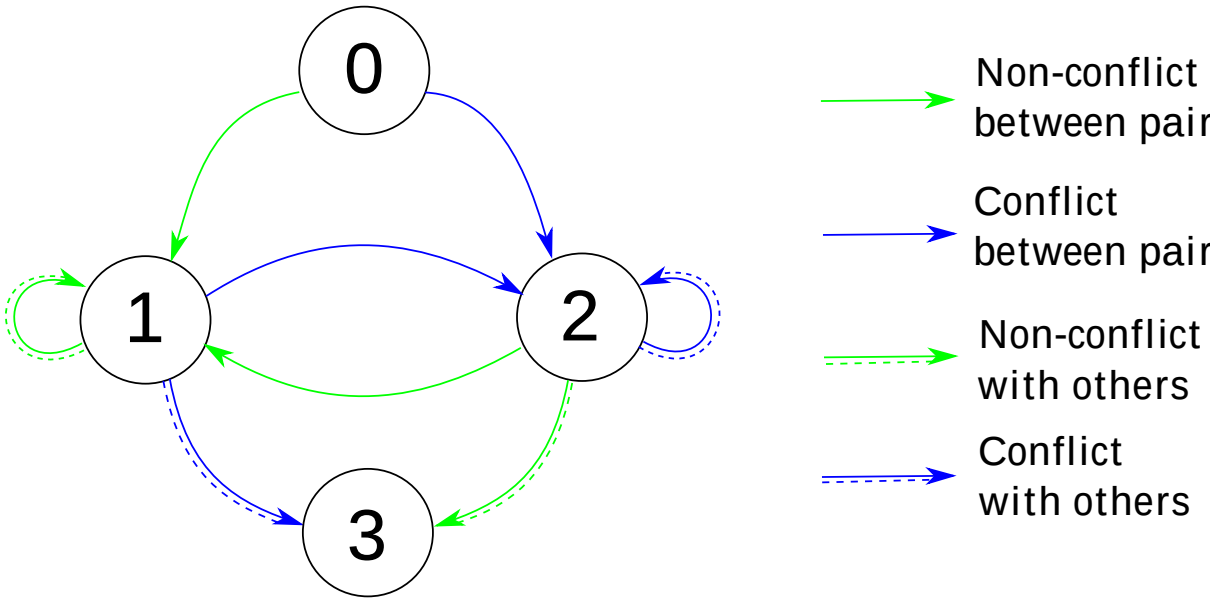

Figure 2.7: Hypothetical state-transition model of conflict for pair of users; state 0 signifies starting of engagement between a hypothetical user pair.

identify three different types of engagement patterns in user clusters:

- **Type-I** clusters tend to be formed with non-conflicting engagement between users. Users in these clusters do not seem to get engaged in conflicting manner with users in other clusters as well.
- **Type-II** clusters are formed with users having mutual conflict. They tend to have conflicting interactions with other clusters as well.
- **Type-III** clusters show a organization-like behavior. These users maintain almost non-conflicting engagement with each other, but aggressive towards other clusters (mostly green regions inside the cluster and blue ones outwards in Figure 2.6).

Type-III clusters tend to grow most compared to type-I and type-II clusters. Different type-III clusters have most inter-cluster conflicts, even greater than that of type-II clusters. Type-I clusters show least growth rate among three types, signifying that these users are less prone to go out of their 'comfort zone'.

This cluster types are of course not completely rigid. Although there is no sign of conversion between type-I and II, both of them can slowly convert into type-III. It is intriguing to observe two different patterns in the formation of type-III cluster – (i) Some of them emerge as type-III from the beginning. Users having no previous engagement form non-conflicting connections with each other. This may signify a probable community interaction among them beyond the discussion platform such as organized campaigners, small group of people using multiple fake user accounts *aka* sockpuppets [31], people are accustomed to each other in real life and sharing similar opinions, etc. (ii) Some of them started as type-I or II and slowly get converted into type-III, which possibly signifies the evolution of engagement via predominant platform interaction.

Users in type-II clusters start changing opinion towards each other with long term interaction and get converted into type-III. Similarly, type-I users tend to start interacting with opposite opinions and convert themselves into type-III. We observe that 33% of the type-III clusters at the end of time frame are the ones converted from type-II, whereas 48% are from type-I. Rest of them started growing as type-III clusters.

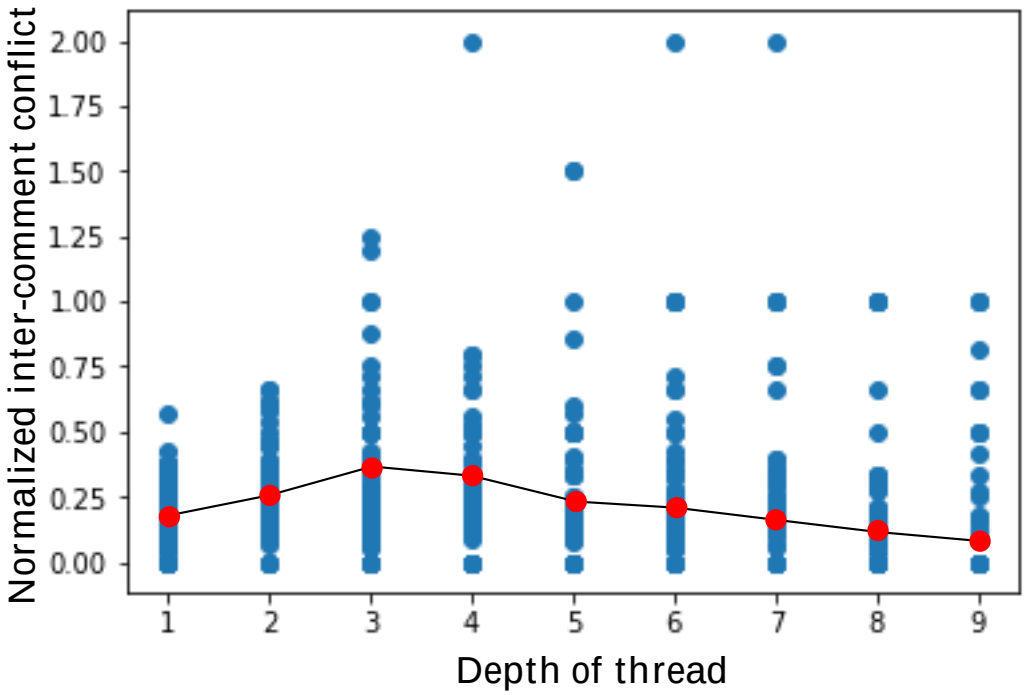

Figure 2.8: Variation of inter-comment conflict with depth of comments in discussion tree.

Formation and evolution of these clusters closely follow the abstract model of user engagement in Figure 2.7. A repeated transition from state 1 along the self loop results in type-I cluster, whereas the same happening on state 2 will result in a type-II cluster. If all the user pairs from state 1 start conflicting with each other, it will lead to a transition to state 2, which implies that a type-I cluster is transformed into type-II. This can only be possible hypothetically; however we did not find any such evidence in our dataset. Likewise, transition from state 1 or 2 to state 3 signifies preferential conflict, resembling type-III clusters.

In Figure 2.8, we plot the variation of inter-comment conflict with the depth of the comments in discussion thread tree. We normalize conflict scores to $(0, 2)$ interval. For comment pairs at depth i and $i + 1$, we plot their conflict at depth i. As it is evident from the plot, a discussion thread is most prone to conflict at depth levels 3 and 4. For interactions at more depth, variance goes up substantially, but average inter-conflict score drops steadily.

Table 2.5 shows an example statistics of different news sources regarding which discussions lead to user clusters. We report this for three different instances $G'(t_1)$, $G'(t_2)$, and $G'(t_3)$ at time t_1, t_2, and t_3 respectively. We take the discussions initiated within past 24 hours for each instance of the network and map the users in each of the largest three clusters to those discussions. As each discussion is related to a news source, this finally maps news sources to clusters. As we can

Cluster index ranked along size	Instance 1	Instance 2	Instance 3
1	Baltimore News (40.02%) Wichita Eagle (25.92%) National Geographic (20.00%) Fox News (13.33%)	Comic Book (76.48%) Wichita Eagle (11.89%) Fox News (1.91%) Detroit News (0.54%)	New York Times (49.03%) Fox News (25.08%) abc13 (25.88%)
2	Baltimore News (100%)	Wichita Eagle (50.57%) Baltimore News (47.12%) National Geographic (2.29%)	BBC (78.52%) Independent (18.61%) New York Times (2.87%)
3	abc13 (44.44%) Fox News (27.78%) Baltimore News (22.24%)	Baltimore News (100%)	Guardian (42.98%) Independent (41.32%) Detroit News (15.70%)

Table 2.5: Percentage of di□erent news sources in user clusters of user-user engagement network. We show the statistics of three largest clusters at three di□erent instances of the network. Up to top four news sources (according to %-contribution) is shown.

see in Table 2.5, there are several common news sources present in first and second instances, whereas almost no common sources is found in the third instance.

Chapter 3

Polarity Prediction

3.1 Introduction

We all use social media more or less almost every day and we noticed that related political discussion on social media is not uncommon. Social media discussion platforms like Twitter[1] or Reddit[2] frequent users often express their views regarding various topics(e.g.- politics, movies etc.). We consider an *entity* as the topic user is expressing views on. In this section, we attempt to predict how a user expresses their views the polarity for a particular entity and attempt to predict the polarity of that user for an unknown entity. Let us begin with an example.

Example: Consider two politicians, Theresa May and Jeremy Corbyn, leaders of the two largest political parties (the Conservative Party and the Labour Party respectively) in the UK. We would expect a regular user to have either (i) opposite sentiments towards them, i.e., positive sentiment towards Theresa May and negative sentiment towards Jeremy Corbyn or vice versa (in case the user supports one of the parties), or (ii) a negative sentiment towards both the politicians (in case the user does not support either of the parties). Since the politicians are rivals, a normal user is unlikely to have a positive sentiment towards both of them. A user having sentiment towards some entities of one of the parties will help us to predict the sentiment of that user for some unknown entities or political party.

More formally, if a user u has polarity for n number of entities which user already expressed through the social media views, based on those polarities we try to predict what can be the polarity for an unknown entity e' of that same user u.

Based on this intuitive definition, we propose two different approaches for polarity detection. The polarities informing the polarity detector are produced by a suitable Target Dependent Sentiment Classification (TDSC) method. Here we pick the two best methods, TDParse [52] and MTTDSC [26], based on earlier experiments done by the Gupta et al. [26].

[1] www.twitter.com
[2] www.reddit.com

3.2 Dataset

3.2.1 Data Collection

Gupta et al. [26] introduced a new TDSC dataset, based on UK election tweets. They first curated a list of candidate hashtags related to the UK General Elections, such as #GE2017, #GeneralElection and #VoteLabour. Using Twitter's streaming API, they collected tweets which contain at least one of these hashtags. The collection was done during a period of 12 days, from June 2, 2017, through June 14, 2017. After removing retweets and duplicates (tweets with the same text), they ended up with 563,812 tweets. All non-ASCII characters were removed from the tweets. Tweets were tagged for named entities using a NER model trained specifically for tweets, developed by [45]. After running the NE tagger, they observed that 158,978 tweets (28.19%) had at least one named entity, 38,809 tweets (6.88%) had at least two named entities, and the remaining 7,992 tweets (1.42%) had three or more named entities. They took all the tweets which had at least two named entities, and randomly sampled an equal number of tweets from the set of tweets which had only one named entity.

3.2.2 Data Filtering

We used the following steps to filter the data: (i) we removed all users (and their tweets) who had posted fewer than 100 tweets during the chosen period. (ii) We inspected the most frequent 2000 entities in the US Election tweet corpus and manually collected 157 entities which were either parties or politicians in the UK. This step was intended to remove tagged entities not related to UK elections. After filtering, we were left with 324,905 ⟨user, entity, tweet⟩ tuples. (iii) Later we also removed tweets, which do not contain any one of the entities, as those tweets do not contribute to the experiments, so we decided to remove them altogether. (iv) After applying previously mentioned filtering methods, we did one round of filtering of the data, where for each individual user u_i, we checked the number of unique tweets made by that user should be at least 20, otherwise; it will be difficult to predict and evaluate the model based on the outcome. Section 3.1 shows the comparative study between all tweets and valid tweets after applying different filtering methods.

For the training and testing purposes, we made sure to split the data using stratified sampling [39]. We made the split in 80:20 ratio.

3.2.3 Data Representation

By running one of the chosen TDSC systems on each surviving tweet, we then generated ⟨user, entity, sentiment⟩ tuples. From these tuples, we generated sentiment probability tuple score for each tweet ⟨pos, neut, neg⟩ and finally, we labeled that tweet belonging to one of the classes based on the probability score (pos:+1, neut:0, neg:-1). We also observed that some users having multiple different polarities for the same entity, to resolve the conflict, in this case, we take the

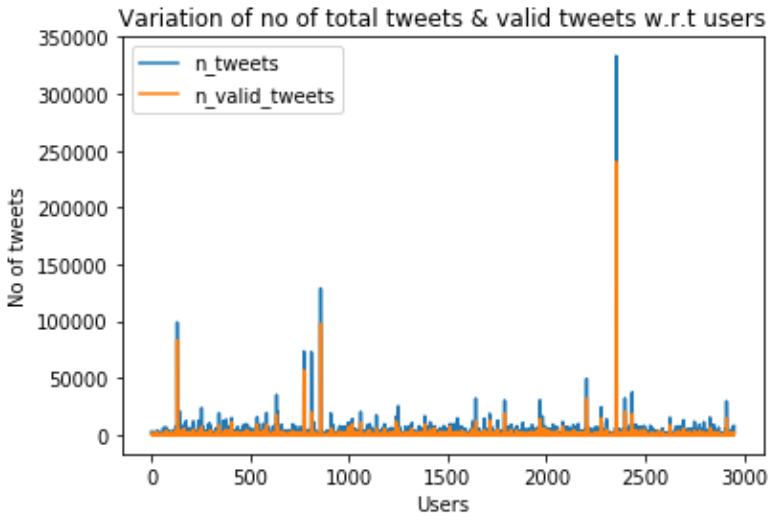

Figure 3.1: No of initial tweets vs. no of filtered tweets of each users.

mean value of the all the polarities expressed by that user and consider the mean value as the polarity score for a specific user towards an entity. More formally if there exists n no of tuples for a specific user u_i towards a specific entity e_j, then we will use $\langle\ u_i,\ e_j,\ s'\ \rangle$, where s' can be formulated using following formula:

$$s' = \frac{1}{n}\sum_{i=1}^{n} s_i \tag{3.1}$$

, where s_i is the sentiment score extracted from the TDSC models for the ith tweet of the user u_i, the reason behind this formulation is if a user has multiple tweets with positive polarity and very few tweets with negative polarity for entity e_j, it is highly likely that the user u_i condemns a strong polarity for the entity e_j.

3.3 Polarity Detection

Recommender systems use the opinions of a community of users to help individuals in that community more effectively identify the content of interest from a potentially overwhelming set of choices [44]. One of the most successful technologies for recommender systems, called *collaborative filtering(CF)*, has been developed and improved over the past decade to the point where a wide variety of algorithms exist for generating recommendations. The underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person. In this polarity prediction task, we use this underlying assumption for the unknown as well as known entities.

For the detection of polarity, we attempt to use different CF based algorithms. We use multiple algorithms to make sure neither data nor TDSC has any dependencies over CF based algorithm. We try out various algorithms and evaluation matrices to evaluate the model, to assess the

robustness of the detection techniques. Following algorithms, we primarily use for the prediction purpose. We explain all the CF techniques briefly in the following sections.

3.3.1 User-user and Entity-entity

In the field of CF, user-user and entity-entity (famously known as item-item) processing are two popular techniques, which primarily use all the data and scores provided by different users or ratings provided for items. [8] and [27] proposed user-user based algorithms, which primarily we refer to as our user-based CF model. For the similarity or correlation measurement, we use Pearson correlation [3], which can be expressed using the following formula:

$$sim(u_x, u_y) = \frac{\sum_{h=1}^{n'}(r_{u_x,i_h} - \overline{r_{u_x}}) - (r_{u_y,i_h} - \overline{r_{u_y}})}{\sqrt{\sum_{h=1}^{n'}(r_{u_x,i_h} - \overline{r_{u_x}})^2}\sqrt{\sum_{h=1}^{n'}(r_{u_y,i_h} - \overline{r_{u_y}})^2}} \tag{3.2}$$

where u_x and u_y indicate xth and yth individual users respectively, r_{u_x} indicates the mean rating given by the user u_x and $r_{i,j}$ is rating provided by the ith user for jth item, for our experiments this will be considered as sentiemtnt class(-1,0 or 1) for the jth entity. Now for the item based method we finally used the following formula for predicting class(or sentiment) of user u_a towards an entity i_a.

$$pred_{u_a,i_a} = \overline{r_{u_a}} + \frac{\sum_{h=1}^{m'} k_{a,h}(r_{u_h,i_a} - \overline{r_{u_b}})}{\sum_{h=1}^{m'} |k_{a,h}|} \tag{3.3}$$

, where $k_{a,h}$ value is same as similarity as mentioned in 3.2.

Similar to the item based model, one of the popular algorithms is item-based CF model - [41], [49] are few popular algorithms of the item-based method. In this case, to predict the rating of the active user using non-active user data, we use item-based data. So, for the prediction of the active entity i_a, we will use the rest of the entities, and it's corresponding data for the final prediction.

$$pred_{u_a,i_a} = \overline{r_{i_a}} + \frac{\sum_{h=1}^{n'} \mu_{a,h}(r_{u_a,i_h} - \overline{r_{u_a}})}{\sum_{h=1}^{n'} |\mu_{a,h}|} \tag{3.4}$$

, where $\mu_{x,y}$ defines weighted factor for yth entity by user xth, which also can be expressed by the following mathematical formula:

$$\mu_{x,y} = \frac{\sum_{h=1}^{m'}(r_{u_h,i_x} - \overline{r_{i_x}}) - (r_{u_h,i_y} - \overline{r_{i_y}})}{\sqrt{\sum_{h=1}^{m'}(r_{u_h,i_x} - \overline{r_{i_x}})^2}\sqrt{\sum_{h=1}^{m'}(r_{u_h,i_y} - \overline{r_{i_y}})^2}} \tag{3.5}$$

3.3.2 K-Nearest Neighbour

In machine learning, the K-nearest neighbors algorithm (K-NN) [13] is a non-parametric method used for classification and regression. We also use K-NN as one of the CF algorithms. In this section, we explain the K-NN algorithms. For these neighborhood-based algorithms, instead of considering the linear combination of users, we need to weight all the user concerning the active user. If user u_i and u_j have a positive sentiment towards and entity e_a, then for the active user u_a is very likely to have a positive sentiment, if u_i and u_j have historically proven themselves as the valuable recommenders. For the *significance weighting* [27] factor, following formula which is known as *Spearman* has been used to calculate the weighted factor. Even though *Spearman* rank correlation is similar to the Pearson, but it does not rely on the model assumptions, computing a measure of the correlation between ranks instead of rating values.

$$\omega_{a,u} = \frac{\sum_{i=1}^{m}(rank_{a,i} - \overline{rank_a}) * (rank_{u,i} - \overline{rank_u})}{\sigma_a * \sigma_u} \tag{3.6}$$

The main drawback for *significance weighting* is that it treats each of the entities evenly for a user to user correlation, but this can not be true always. E.g., it was noticed that, for an entity e', which was not a part of the UK election, most of the users had a neutral label or class label as 1. So if two users have neutral sentiment towards an outsider entity, this information tells very little about shared political interests between those two users. But if a user has a positive sentiment towards *Theresa May*, then it is very likely they support *the Conservative Party* and all the entities supporting that particular party. We hypothesize that giving the explicit weight to the distinguishing entities would improve the performance. We incorporate entity-variance weight factor to the Pearson correlation. By incorporating an explicit weight term, it increases the influence of the entities of low variance. The new correlation can be expressed as the following formula mathematically:

$$\omega_{a,u} = \frac{\sum_{i=1}^{m} v_i * z_{a,i} * z_{u,i}}{\sum_{i=1}^{m} v_i} \tag{3.7}$$

3.3.3 Matrix Factorization

Matrix factorization based recommendation methods gains great success due to their efficiency and accuracy. Let us assume that we have M no of users and N no of entities. Let us assume matrix S denotes the sentiment matrix, where S_{ij} denotes the sentiment of ith user towards jth entity. Let $U \in \mathbb{R}^{MxD}$ and $V \in \mathbb{R}^{NxD}$ is the user and entity latent feature matrices, two vectors U_i and V_j represent the user-specific and entity-specific latent feature vectors. D is the dimension of the user feature vector and item feature vector, which is much less than M and N. In probabilistic matrix factorization (PMF) [36] method, the conditional probability of observed sentiment matrix S is modeled as:

$$p(R|U,V,\sigma^2) = \prod_{i=1}^{M}\prod_{j=1}^{N}[N(R_{ij}|U_iV_j^T,\sigma^2)]^{I_{ij}} \tag{3.8}$$

For more details please refer to the actual paper.

[42] extended the matrix factorization to the Biased Matrix Factorization(BMF), in which the predicted rating is the sum of the inner product of user feature vector and item feature vector and the bias values. The way of learning the model of biased matrix factorization is very similar to the method of determining the model without bias values. We will be using BMF, as our last model for the prediction.

3.4 Results and Analysis

We perform all the above-mentioned CF-based algorithms with the extracted polarity from both the TDSC models(TDParse and MTTDSC). As shown in table 3.1, we can see BMF seems to outperform all the other algorithms.

	Model	**Accuracy**	**RMSE**	**Precision**	**Recall**	**F1**
TDParse	**User-User**	47.3%	2.897	45.23	41.90	43.50
	Item-Item	41.1%	3.230	38.67	33.81	36.08
	K-NN-Significance	42.0%	3.232	39.73	36.87	38.25
	K-NN-Variance	70.7%	1.314	63.25	59.18	61.15
	BMF	81.2%	**0.514**	78.39	76.46	77.41
MTTDSC	**User-User**	49.2%	2.808	43.28	42.81	43.04
	Item-Item	39.7%	3.251	37.28	33.91	35.52
	K-NN-Significance	40.7%	3.255	38.31	35.05	36.61
	K-NN-Variance	71.6%	1.376	69.83	65.28	67.48
	BMF	**84.3%**	0.64	**81.29**	**78.07**	**79.64**

Table 3.1: Comparison of di□erent CF algorithms for di□erent TDSC algorithms of Polarity Prediction

Table 3.1 shows the results of all algorithms concerning both of the TDSC algorithm. Although Root-Mean-Square-Error(RMSE) of the TDParse-BMF performs better than compared to other algorithms, on the metric of accuracy MTTDSC-BMF seems to work much better on the prediction of the polarity. By exploiting the easier auxiliary task of whole-passage sentiment classification, MTTDSC improves on other TDSC baselines for this task. The auxiliary LSTM learns to identify corpus-specific, position-independent sentiment in words and phrases, whereas the main LSTM learns how to associate these sentiments with designated targets.

Chapter 4

Temporal Behaviour Prediction

4.1 Introduction

Of all the people who use social media platforms frequently, we must have seen some some posts by the similar user where that user was a massive supporter of a particular political party or any sports person or any movie star, by with a specific span of time that user changes the support and starts protesting against it due to some circumstantial situations or due to the disappointment of some action by that entity(indicating same reference as we discussed in Chap 3) or maybe that user was not a core supporter from the very beginning itself. This uncertain temporal polarity behavior of a user can help us to detect the collusive users. This type of temporal actions of individual users, who can be a potential influencer or a role model can guide a considerable part of the social media user base to the wrong path.

There has been a lot of work with the time series data or temporal data related problems. Stock market pricing is the most common example of time-series data. The research community has made a significant effort on the analysis of the data and behavior as well as prediction of the stock market [7], [50]. Our primary target in this section is very similar to this task. We aim to predict the polarity of a user towards a particular entity based on all the polarities in previous time series windows.

More formally, for an active user u_a, if the user is frequent about an entity e_a and also tweets about that entity very containing strong polarities in the previous tweets, our task is to analyze how the sentiment of e_a changes with time for u_a. We divide the complete time frame with fifteen days non-overlapping time frame windows and try to predict sentiment(or polarity, for this task, these terms are interchangeable) for the next time frame. Same as mentioned in Chap 3, we use TDParse and MTTDSC to create $\langle$ user, entity, sentiment $\rangle$ tuple in each time frame.

4.2 Dataset

The dataset for this task is the same as the one we mentioned in chapter 3. As mentioned in the previous chapter, data collection is the same as the previously mentioned sections. We follow steps mentioned in 3.2.1 to collect the data. For the filtering purpose for each user, we take a window of fifteen consecutive days of the time frame and represent that data by the previously mentioned process, which is mentioned in 3.2.3. So for each time frame, we have ⟨user, entity, sentiment⟩. If an active user u_a has no tweets containing active entity e_j, then tuple containing sentiment score is the same as the last time frame sentiment score. We also see some of the users can have an initial tweet for an entity to the latter part of the time frame, in that case, we consider that user has neutral sentiment for that particular entity.

4.3 Methodology

For this prediction task also, we use multiple architectures to check the robustness of the TDSC algorithms. In recent years, deep learning has emerged as one of the most popular machine learning techniques, yielding state-of-the-art results for a range of supervised and unsupervised tasks. The primary reason for the success of deep learning is its ability to learn high-level representations which are relevant for the task at hand. These representations are learned automatically from data with little or no need for manual feature engineering and domain expertise. For sequential and temporal data, Long short-term memory (LSTM) [11] recurrent neural networks (RNNs) have become the deep learning models of choice because of their ability to learn long-range patterns.

For the first architecture, we use vanilla LSTM network for the learning long-range dependencies and prediction of the label value for the last window.

For the second model, we use two popular individual DL-based architectures - Encoder and Gated Recurrent Unit(GRU), a flavor of vanilla Recurrent Neural Network(RNN). Initial feature vector has been passed through the encoder architecture, the sole purpose of this network is to enforce the model to help the model to learn some meaning-full features, which will later help upcoming GRU network to determine the data distribution and predict the polarity of the next time frame.

For the third model, we also use another DL-based architecture proposed by [55]. Yang et al. proposed a hierarchical attention network(HAN) primarily focused on document classification. HAN progressively builds a document vector by aggregating important words into sentence vectors and then aggregating important sentences vectors to document vectors. With the regular feature set to the model, we also incorporate encoded sentiment of the last time frame to create a new feature set. This new feature set will be passed through the network for the classification of the sentiment of the current time frame.

4.4 Results and Analysis

	Model	Accuracy	Precision	Recall	F1
TDPARSE	**LSTM**	51.97%	48.76	46.02	47.35
	Encoder+GRU	55.21%	54.68	53.79	54.23
	HAN	69.4%	64.29	62.31	63.28
MTTDSC	**LSTM**	53.26%	49.17	47.94	48.55
	Encoder+GRU	58.68%	53.92	51.27	52.56
	HAN	**71.03%**	**69.72**	**68.69**	**69.20**

Table 4.1: Comparison of di□erent algorithms for di□erent TDSC algorithms for temporal polarity prediction

Table 4.1 shows that HAN+MTTDSC seems to outperform all the other model+TDSC combinations due to the enriching combination of both of the DL architectures. The auxiliary LSTM of MTTDSC learns to identify corpus-specific, position-independent sentiment in words and phrases, whereas the main LSTM learns how to associate these sentiments with designated targets and attention based architecture able to focus on the target words, which helps to predict the polarity of that user in the given time frame.

Chapter 5

Conclusions and Future Work

5.1 Conclusion

Our analyses in provide novel insights into the conflict dynamics over large-scale online discussion. We show how different news sources get different reactions from their audience and how this varies temporally. We identified three distinct types of user clusters developed in Reddit *r/news* community, based on the attitude towards other users and engagement patterns. We also provided a hypothetical state-transition model of user engagement, which is closely followed by actual interaction patterns.

5.2 Future Work

As we observed that the experimental results in Chap 4 is not extraordinary. The primary probable reason of this can be dataset we have used. The time frame window we used for 15 days, but majority of that user did not have much tweets with such short time slice window very frequently, so the model could not learn the underlying pattern of each user in for each entity in the given current time frame. Also we might can use some DL-based architecture which can probably leverage the small amount of data we have. Better solution to tackle this problem would be to get much richer dataset, in which users are much frequent and active for the selected time slice window.

Bibliography

[1] Luca Maria Aiello, Alain Barrat, Rossano Schifanella, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. Friendship prediction and homophily in social media. *ACM Transactions on the Web (TWEB)*, 6(2):9–29, 2012.

[2] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: an open source software for exploring and manipulating networks. In *ICWSM*, pages 11–20, 2009.

[3] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[4] Rianne van den Berg, Thomas N Kipf, and Max Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.

[5] Carl-Hugo Björnsson. Readability of newspapers in 11 languages. *Reading Research Quarterly*, pages 480–497, 1983.

[6] Catherine A Bliss, Morgan R Frank, Christopher M Danforth, and Peter Sheridan Dodds. An evolutionary algorithm approach to link prediction in dynamic social networks. *Journal of Computational Science*, 5(5):750–764, 2014.

[7] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.

[8] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[9] Erik Cambria, Soujanya Poria, Devamanyu Hazarika, and Kenneth Kwok. Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In *AAAI*, pages 1–10, 2018.

[10] Tanmoy Chakraborty, Ayushi Dalmia, Animesh Mukherjee, and Niloy Ganguly. Metrics for community analysis: A survey. *ACM Computing Surveys (CSUR)*, 50(4):54, 2017.

[11] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[12] Michael D Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on twitter. In *ICWSM*, pages 1–10, 2011.

[13] Thomas M Cover, Peter E Hart, et al. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[14] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pages 3844–3852, 2016.

[15] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *ACL*, volume 2, pages 49–54, 2014.

[16] Subhabrata Dutta, Tanmoy Chakraborty, and Dipankar Das. How did the discussion go: Discourse act classification in social media conversations. In *Linking and Mining Heterogeneous and Multi-view Data*, pages 137–160. Springer, 2019.

[17] Kelwin Fernandes, Pedro Vinagre, and Paulo Cortez. A proactive intelligent decision support system for predicting the popularity of online news. In *Portuguese Conference on Artificial Intelligence*, pages 535–546. Springer, 2015.

[18] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[19] Thomas MJ Fruchterman and Edward M Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

[20] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Quantifying controversy on social media. *ACM Transactions on Social Computing*, 1(1):3, 2018.

[21] Venkata Rama Kiran Garimella and Ingmar Weber. A long-term analysis of polarization on twitter. In *ICWSM*, pages 1–10, 2017.

[22] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *SIGCHI*, pages 211–220. ACM, 2009.

[23] Pedro Calais Guerra, Wagner Meira Jr, Claire Cardie, and Robert Kleinberg. A measure of polarization on social media networks based on community boundaries. In *ICWSM*, pages 1–10, 2013.

[24] Robert Gunning. The fog index after twenty years. *Journal of Business Communication*, 6(2):3–13, 1969.

[25] Divam Gupta, Tanmoy Chakraborty, and Soumen Chakrabarti. Girnet: Interleaved multitask recurrent state sequence models. *arXiv preprint arXiv:1811.11456*, 2018.

[26] Divam Gupta, Kushagra Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. Multi-task learning for target-dependent sentiment classification. *arXiv preprint arXiv:1902.02930*, 2019.

[27] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999*, pages 230–237. Association for Computing Machinery, Inc, 1999.

[28] Julie Kauffman, Aristotelis Kittas, Laura Bennett, and Sophia Tsoka. Dyconet: a gephi plugin for community detection in dynamic complex networks. *PloS one*, 9(7):e101357, 2014.

[29] Yaser Keneshloo, Shuguang Wang, Eui-Hong Han, and Naren Ramakrishnan. Predicting the popularity of news articles. In *SDM*, pages 441–449. SIAM, 2016.

[30] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[31] Srijan Kumar, Justin Cheng, Jure Leskovec, and V.S. Subrahmanian. An army of me: Sockpuppets in online discussion communities. In *WWW*, pages 857–866, 2017.

[32] Srijan Kumar, William L Hamilton, Jure Leskovec, and Dan Jurafsky. Community interaction and conflict on the web. In *WWW*, pages 933–943. International World Wide Web Conferences Steering Committee, 2018.

[33] Stephen J Lepore. Social conflict, social support, and psychological distress: evidence of cross-domain buffering effects. *Journal of personality and social psychology*, 63(5):857, 1992.

[34] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[35] Michal Lukasik, PK Srijith, Duy Vu, Kalina Bontcheva, Arkaitz Zubiaga, and Trevor Cohn. Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In *ACL*, volume 2, pages 393–398, 2016.

[36] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

[37] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. Semeval-2016 task 6: Detecting stance in tweets. In *SemEval*, pages 31–41, 2016.

[38] Reed E Nelson. The strength of strong ties: Social networks and intergroup conflict in organizations. *Academy of Management Journal*, 32(2):377–401, 1989.

[39] Jerzy Neyman. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97(4):558–625, 1934.

[40] Elizabeth Levy Paluck, Hana Shepherd, and Peter M Aronow. Changing climates of conflict: A social network experiment in 56 schools. *PNAS*, 113(3):566–571, 2016.

[41] Manos Papagelis and Dimitris Plexousakis. Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. *Engineering Applications of Artificial Intelligence*, 18(7):781–789, 2005.

[42] Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8, 2007.

[43] Alicja Piotrkowicz, Vania Dimitrova, Jahna Otterbacher, and Katja Markert. Headlines matter: Using headlines to predict the popularity of news articles on twitter and facebook. In *ICWSM*, pages 1–10, 2017.

[44] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–59, 1997.

[45] Alan Ritter, Sam Clark, Oren Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

[46] Georgios Rizos, Symeon Papadopoulos, and Yiannis Kompatsiaris. Predicting news popularity by mining online discussions. In *WWW*, pages 737–742. International World Wide Web Conferences Steering Committee, 2016.

[47] Sara Rosenthal and Kathy McKeown. I couldn't agree more: The role of conversational structure in agreement and disagreement detection in online discussions. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 168–177, 2015.

[48] Niek J Sanders. Sanders-twitter sentiment corpus. *Sanders Analytics LLC*, 242, 2011.

[49] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. Item-based collaborative filtering recommendation algorithms. *Www*, 1:285–295, 2001.

[50] Shunrong Shen, Haomiao Jiang, and Tongda Zhang. Stock market forecasting using machine learning algorithms. *Department of Electrical Engineering, Stanford University, Stanford, CA*, pages 1–5, 2012.

[51] Robert Speer and Joshua Chin. An ensemble method to produce high-quality word embeddings. *arXiv preprint arXiv:1604.01692*, 2016.

[52] Bo Wang, Maria Liakata, Arkaitz Zubiaga, and Rob Procter. Tdparse: Multi-target-specific sentiment recognition on twitter. In *EMNLP*, pages 483–493, 2017.

[53] Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, 58(1):1–38, 2015.

[54] Bo Wu and Haiying Shen. Analyzing and predicting news popularity on twitter. *International Journal of Information Management*, 35(6):702–711, 2015.

[55] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[56] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

[57] Amy Zhang, Bryan Culbertson, and Praveen Paritosh. Characterizing online discussion using coarse discourse sequences. 2017.

[58] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *NIPS*, pages 5165–5175, 2018.

[59] Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, Michal Lukasik, Kalina Bontcheva, Trevor Cohn, and Isabelle Augenstein. Discourse-aware rumour stance classification in social media using sequential classifiers. *Information Processing & Management*, 54(2):273–290, 2018.

Appendix A

List of Publications by the Candidate

Following is a list of all the publications by the candidate including those on and related to the work presented in the thesis. The publications are arranged in chronological order.

A.1 Conferences

1. **A. Mukherjee**, S. Tiwari, T. Chowdhury, T. Chakraborty. "Automatic Curation of Content Tables for Educational Videos" In The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, July 21-25, 2019, Paris, France. (Accepted)

2. S. Dutta, G. Kaur, S. Mongia, **A. Mukherjee**, T. Chakraborty, D. Das. "Into the Battlefield: Quantifying and Modeling Intra-community Conflicts in Online Discussion" In 28th ACM International Conference on Information and Knowledge Management (CIKM), November 3rd-7th, 2019, Beijing, China. (Under Review).