Generating Influential Replies for Tweets on a Political Topic

Ashwin Sudhir, Chethan Kumar Kolar Ananda Kumar, Nidhal Selmi,
Shivam Dhar, Shubham Verma, Shubham Vipul Majmudar

Computer Science and Engineering,
Arizona State University, Tempe, AZ 85281, USA
{asudhir1, ckolaran, nselmi, sdhar3, sverma41, smajmuda}@asu.edu

***Abstract*—In the era of social networking, it is easy to influence people or reach out to a target audience with a new product or service. This can lead to the spread of positive as well as negative elements in the network depending on the type and source of information. Given a piece of information, it becomes essential to understand how different people would react to it. The objective of this project is to build a topic-based language model for a retweet network. We predict the reply of a user to a tweet, based on their political inclinations.**

***Index Terms*—retweet network, text generation models, retrieval models, deep learning, neural networks, LSTM**

I. Introduction

On the micro-blogging Twitter network, users can support or oppose a particular tweet by replying or retweeting. This forms two communities within the tweeters — one of supporters of a topic and others against it based on their affiliations. This leads to interest in researching models for predicting these replies for a given tweet.

We can also study the shift of views for these tweeters given episodes of twitter data for a specific user. This will help in identifying patterns of user support for a topic over a period of time.

Thus, topic-based tweet reply generation supervised model will be a result of optimal learning based on the user score and tweets previously posted by the user on the topic. Section 3 of the report deals with the related work carried out in this direction. Section 4 describes the system architecture and algorithms developed in order to build a reply generation system. Section 5 provides a detailed explanation on the dataset used and the preprocessing steps performed to prepare the dataset for further use. Section 6 emphasizes on the observations made and the results achieved. Section 7 highlights the responsibilities shared by the teammates in order to execute the project efficiently and section 8 concludes with the overall experience and future scope of the project.

II. Problem Statement

Given a tweet in a political context, the objective is to generate a reply which is either for or against the tweet i.e. left or right aligned based on the tweeters political inclination.

III. Related Work

A. Redford et al [1] offers a novel language mode learner that can achieve baseline performance on multiple tasks in a zero-shot setting. This highlights that language models can be deployed in micro-blogging setting if fine tuned on different political views.

Social influence can be studied from a topic modeling perspective as explained by Barbieri, N et al [2]. They introduce novel topic-aware influence-driven propagation models that experimentally result to be more accurate in describing real-world cascades than the standard propagation models studied like Independent Cascade and Linear Threshold models.

Hanghang Tong et al [3] explain the mechanism of gelling and melting for large graphs by edge manipulation i.e. gelling refers to the study of the network on the addition of edges whereas melting refers to the study of the network on the removal of certain edges.

J. Leskovec et al [4] explains how old blogs cite and influence each other, how these links evolve and how the popularity of these blogs drop after some time. The temporal and topological aspects of the old blog network were studied like the common shapes of the information cascades, their size, and distribution, network characteristics like periodicity - bursty or periodic, etc.

Deep Neural Networks are powerful models that have achieved excellent performance on difficult learning tasks. They work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. Ilya et al [5] present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. The method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector.

J. Leskovec et al [6] explains how old blogs cite and influence each other, how these links evolve and how the popularity of these blogs drop after some time. The temporal and topological aspects of the old blog network were studied like the common shapes of the information cascades, their

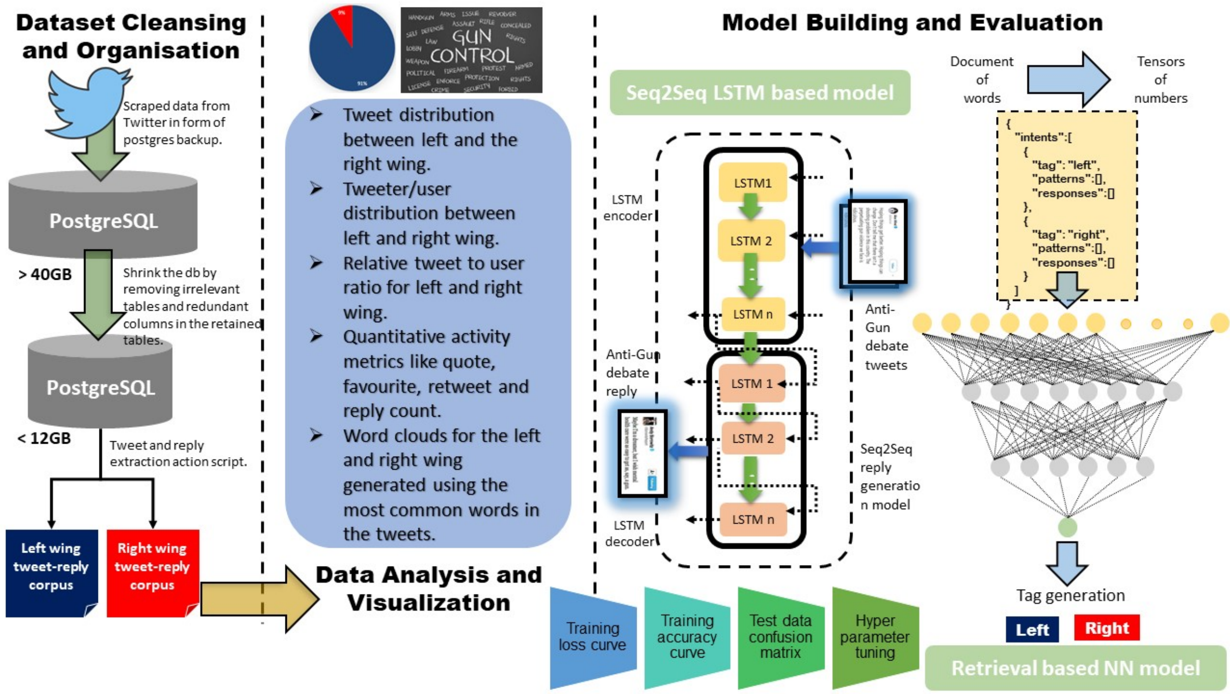

Fig. 1. System architecture for reply generation system

size, and distribution, network characteristics like periodicity — bursty or periodic, etc.

IV. System Architecture and Algorithms

A. Architecture Design

The reply generation system consists of following modules:

1) Dataset cleaning and organization: It involves data cleansing like reducing the size of data by removing unwanted objects from the database, extracting the tweet-reply data and making data available to be consumed by the models we build. It can be either in form of tab separated values with tweet and it's corresponding reply or in form of a JSON structure specifying the tag for each pair of tweet-reply stored.
2) Data analysis and visualization: We analyze different statistical properties in the dataset like tweet count, user count, tweet to user distribution, quote count, reply count, retweet count and favorite count for the two wings - left and right and visualize the same through different plots and wordclouds.
3) Model building and evaluation: We take two approaches for building ours reply generation model - seq2seq LSTM based generative model and retrieval model. We evaluate the results of both the models and report the model metrics.

B. Algorithms

We explore two approaches for reply generation - generative approach and retrieval based. For the generative model, we used Seq2Seq neural network based text generation model. It consists of two parts, an encoder and a decoder, both of which are stacked LSTM layers. The encoder maps a variable-length source sequence to a fixed-length vector, and the decoder maps the vector representation back to a variable-length target sequence. The two networks are trained jointly to maximize the conditional probability of the target reply given a source tweet.

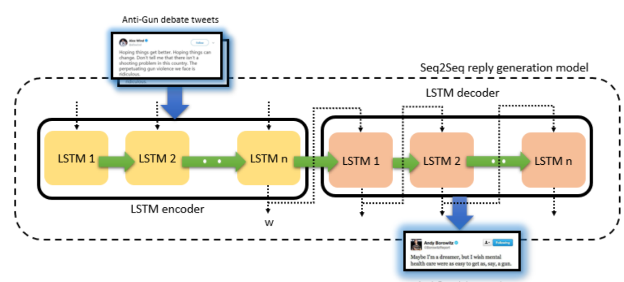

Fig. 2. Example of an Anti-Gun topic based reply generation model.

We build two models — one using tweets which are anti topic and another using tweets for the topic. Fig. 2, shows a sample model for Anti-gun debate reply generation model. The entire tweet dataset corresponding to one of the ideologies is given as input to the encoder. During the training phase, the actual reply received for the original tweet is used for building a strong model, which continuously improves the objective function based on cosine similarity between the reply generated and actual reply. The model learns the context and the words used by the tweeters of a particular ideology. We also choose parameters like the number of LSTM layers, the size of each LSTM layer, batch size, learning rate, vocabulary size for the encoder and decoder and some checkpoints.

For the retrieval based model, we transform the tab separated values for tweet-reply into a JSON structure having tags - left or right and corresponding tweets and replies nested under each of the tags. A neural network is trained using the processed data by converting the documents of words to tensors of numbers. Additionally we write a framework to process tweets and generate replies. The input tweet during the testing

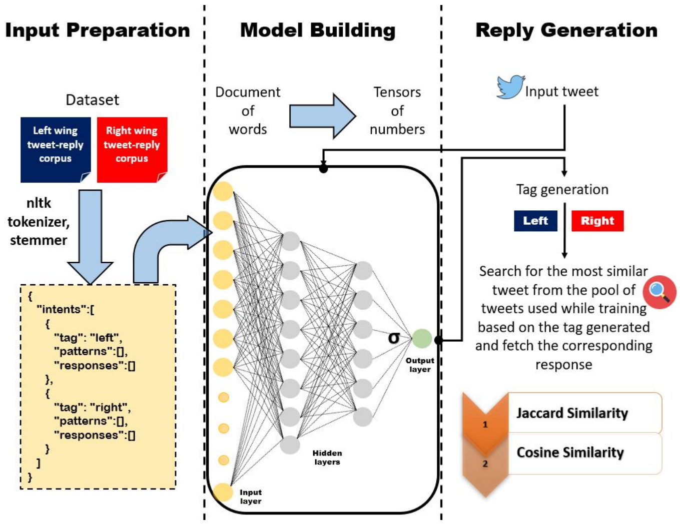

Fig. 3. Design of the retrieval tweet-reply model.

phase is translated into bag of words. The response processor predicts the tag for the tweet, and searches for the most similar tweet from the pool based on cosine similarity and jaccard similarity measures. It finally fetches the corresponding reply for the most similar tweet and returns the same.

V. Dataset

A. Description

We used the twitter dataset scraped for GunDebate topic. The postgresql dump consists of 16 tables, out of which the following were retained and used:

- replyto: gives information about the users who have replied to a tweet by another user.
- tweet: has attributes associated with a tweet like user id, screen name, tweet content, and tweet id.
- userinfo: contains user fields like user id, name, user url, screen name, location, etc.
- userlabel: this relation contains a tag associated for each user — left or right.

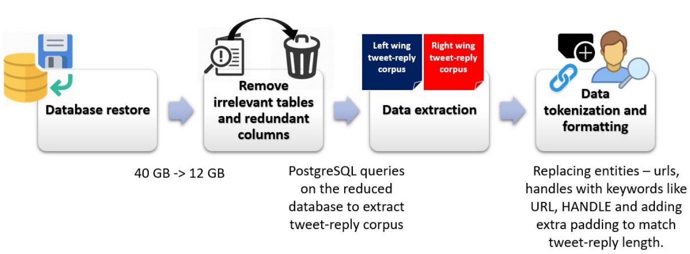

Fig. 4. Steps involved in data preprocessing.

B. Preprocessing

The size of postgresql dump was more than 40GB. It becomes quite difficult to work with such loads of data. In order to make the processing faster, we removed all the redundant and not important objects/tables from the database after restoring the same through psql. Next, we extracted the left and right aligned tweet-reply corpus in form of tab separated values. Finally, we replaced entities like urls, usernames, handles by tokens like URL, HANDLE. Additionally, to ensure same tweet-reply length, we padded the shorter sequence with PAD token.

C. Data Analysis

We find statistical properties in the dataset. Fig 5 shows the tweet and tweeter distribution between the two wings. It also shows the relative tweet-user distribution.

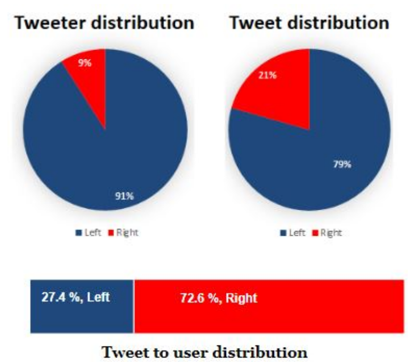

Fig. 5. Tweet and tweeter distributions.

We also plot the quote count, reply count, retweet count and favorite count for the two wings and visualize the same through barplots as shown in Fig 6.

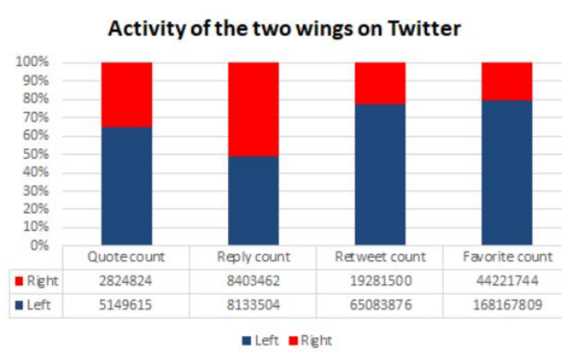

Fig. 6. Plot of statistical properties for the two wings.

To visualize the commonly used words by the tweeters, we draw word clouds for 1000 tweets having users from left and right wing as shown in Fig 7. This uses the principle of term frequency in the documents, in our case tweets and the replies, and arranges the words based on the number of times used by the tweeter with varying font size.

Fig. 7. Word clouds showing top words used by the right and left aligned tweeters.

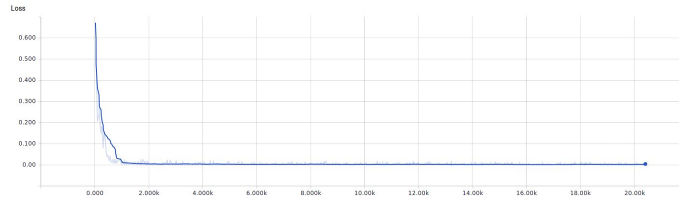

Fig. 10. The loss values for the model during training phase for each epoch.

VI. Evaluation

This section reports the observations and the inferences drawn based on the approaches taken, along with the demo screens built for the interactive version of the system.

A. Generative model

The generative model built can be evaluated by a human expert. The semantics of the generated response is what the expert looks for. We ran the model for 200-250 epochs and tested it for some input tweets. The response generated consisted of only some user handles and Urls.

B. Retrieval model

We report the following metrics for the retrieval based model on 1012 tweets - 98 left inclined tweets and 914 right inclined tweets, with 80-20 training and test split, run for 200 epochs. The plot of accuracy for the model during training phase, the loss curve for the model during training phase, confusion matrix for the test data, the probability plots for generated tags and LDA topic modelling on the input data are further discussed.

Fig. 8. The accuracy values for the model during training phase for each epoch.

	Left	**Right**
Left	TP = 14	FN = 5
Right	FP = 7	TN = 152

**Considering left as true class*

- Precision (Left class) P_l = TP/(TP+FP) = 0.66
- Recall (Left class) R_l = TP/(TP+FN) = 0.73
- Precision (Right class) P_r = TP/(TP+FP) = 0.95
- Recall (Right class) R_r = TP/(TP+FN) = 0.96
- Mean Precision $P = (P_l + P_r)/2 = 0.81$
- Mean Recall $R = (R_l + R_r)/2 = 0.85$
- Accuracy = (TP+TN)/(TP+FP+FN+TN) = 0.93
- F1 score = 2*P*R/(P+R) = 0.83

Fig. 9. The confusion matrix for test data.

The confusion matrix shown in Fig 9 considers left as true class. We find the true positives (TP), false negatives (FN), false positives (FP) and true negatives (TN), which are further used to calculate precision and recall. Similarly, we compute precision and recall considering right as true class. Finally, we report accuracy and F1 score calculating mean precision and recall.

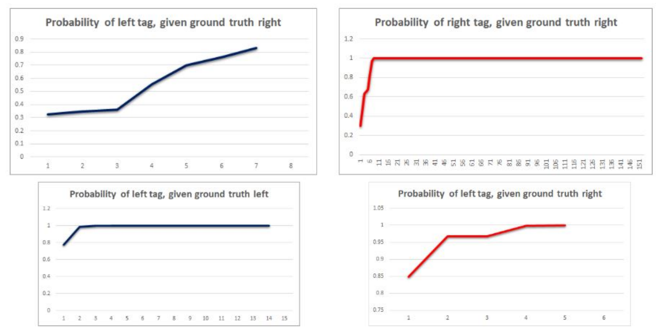

Fig. 11. The probability plots for tag generated by the model.

We plot the probabilities with which the tag is assigned to the input test tweet by the classifier. We visualize this curve for TP, FP, FN and TN as shown in Fig 11.

For improving the quality of retrieval model, we tried topic modelling using LDA on the dataset. The topics generated showed no inherent inclination towards any ideology, which could help adding contextual information about the tweets in the model. The results are shown in Fig 12.

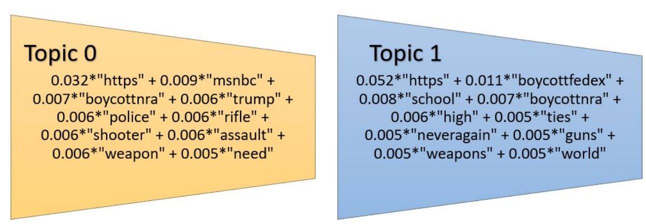

Fig. 12. The words in the two topics generated by LDA.

C. User Interface

We provide a web graphical user interface for users to interact with the backend model. Since, retrieval model gave better results as compared to the generative model, we exposed the retrieval model through RESTful APIs and built the templates in HTML/CSS to take in user input tweet and display the response generated by the model. Fig 13. shows

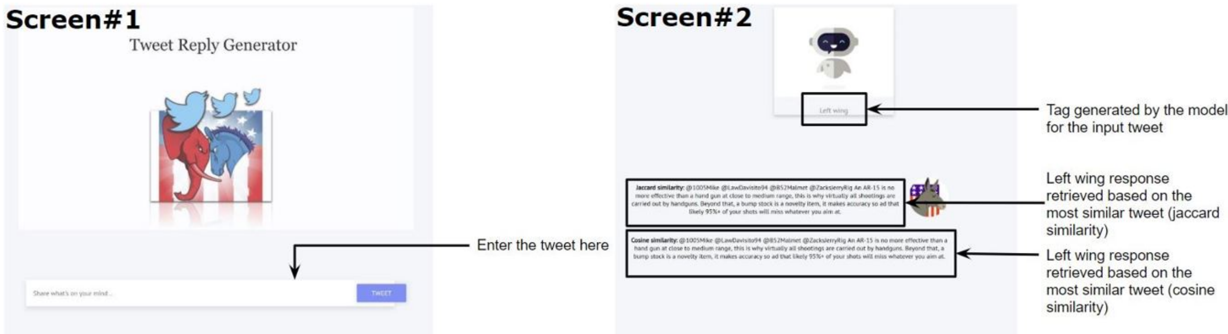

Fig. 13. User interface screens for the retrieval model

the two screens built. Screen 1 allows user to enter the tweet. The model assigns a tag to this tweet and displays it on the next screen below in addition to the most similar response generated based on cosine similarity and jaccard similarity metrics.

VII. Division of work

This section describes the work done by each of the teammates to ensure timely deliverable of the required submissions and executing milestones set for completing each of the tasks. The team used gitlab for maintaining the source code and tracking issues to be worked on.

Chethan and Nidhal worked on the dataset preparation - dataset cleansing, preprocessing and transforming into CSV and JSON. The task was accomplished using python packages like nltk, numpy and regex operations.

Shubham V analyzed various data points through bar plots and word clouds. This involved writing of PostgreSQL query scripts and python programming for visualizations.

Ashwin and Shubham M built a generative model using LSTM based Seq2Seq via python libraries - keras and tensorflow.

Shivam developed the retrieval model using tensorflow library in python and the related user interface using flask, javascript and HTML.

The team also evaluated the models on various metrics and reported the results as shown in section 6.

Member	Task
Ashwin Sudhir	Generative model
Chethan Kumar	Dataset preparation
Nidhal Selmi	Dataset preparation
Shivam Dhar	Retrieval model, GUI
Shubham Verma	Data analysis
Shubham Vipul	Generative model

VIII. Conclusion

The project gave a good exposure on the different techniques one can employ in a real world application related to the semantics of web mining - dataset preparation, tokenization, term frequency in a document, doc representation as tensors of numbers, data visualization, neural network models, model evaluation metrics, etc.

Transforming raw data into meaningful structure - TSV or JSON was the primary task. This was followed by the choice of model for learning replies for a given tweet, which could be later used for politically aligned tweet generation. The models built were evaluated by either human experts or the traditional means of precision, recall, accuracy and f1 score.

The observations made through the two approaches taken and success and failure metrics, both were reported.

As an extension to this, one can further improvise the reply generation system by extending it for multiple topics. This would require a notion of context to be attached with the tweet during training and will enable the tweet bot to reply on any of the tweets posted without any restrictions.

Acknowledgment

We are very thankful to Prof. Hasan Davulcu for his valuable guidance in finalizing the project in the Information Dissemination domain, to Mert Ozer for enabling us with the required dataset, and Amin Salehi for his support. We would like to thank Lincoln Slade, Richard Willis and Don Newhouse for setting up workstations in the lab for our perusal. At last, we would extend our gratitude to all involved directly and indirectly in the project.

References

[1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever. Language Models are Unsupervised Multitask Learners.

[2] Barbieri, N., Bonchi, F., Manco, G.: Topic-aware social influence propagation models. In: ICDM, pp. 8190 (2012).

[3] Hanghang Tong, B. Aditya Prakash, Tina Eliassi-Rad, Michalis Faloutsos, Christos Faloutsos, Gelling, and melting, large graphs by edge manipulation, Proceedings of the 21st ACM international conference on Information and knowledge management, October 29-November 02, 2012, Maui, Hawaii, USA.

[4] Hanghang Tong, B. Aditya Prakash, Charalampos Tsourakakis, Tina Eliassi-Rad, Christos Faloutsos, Duen Horng Chau, On the Vulnerability of Large Graphs, Proceedings of the 2010 IEEE International Conference on Data Mining, p.1091-1096, December 13-17, 2010.

[5] Ilya Sutskever , Oriol Vinyals , Quoc V. Le, Sequence to sequence learning with neural networks, Proceedings of the 27th International Conference on Neural Information Processing Systems, p.3104-3112, December 08-13, 2014, Montreal, Canada.

[6] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs: Patterns and a model. In Society of Applied and Industrial Mathematics: Data Mining (SDM07), 2007.