Using Agglomerative Clustering of Word Embeddings to Categorize Twitter Users

Yasmin Albeny, Lianne Lefsrud*, Eleni Stroulia†, Sheldon Roberts†, Candelario Gutierrez†

†Computing Science, University of Alberta, Edmonton Alberta Canada

*Chemical and Materials Engineering, University of Alberta, Edmonton Alberta Canada

Motivation

Online discussions about public policy are essential to good policy making. By examining public discourse, opinions and perspectives can be acknowledged and considered to find adequate solutions. Such a public policy involves energy policies, more specifically, the Energy East pipeline project announced by TransCanada on August 1, 2013. This project was emblematic, since it became heavily debated on social platforms like Twitter, among multiple groups on economic, political, environmental, and moral grounds. This project aims to categorize the users to identify different demographic's positions on this topic.

Data Processing

The data processing pipeline that was followed and the set of tools used for this project as seen in Figure 2 were the following:

1. Data Collection

Twarc2 is an application used to collect Twitter posts from users' timelines and users' profile description, retrieved from the official Twitter API in CSV format for easier data manipulation.

2. Data Wrangling

Pandas was used to select, filter, and clean users' respective tweets through dataframes to convert such data into words' vector through the GloVe model. By using a words' vector, composed of a direction and a value, words can be compared to one another. See Figure 1.

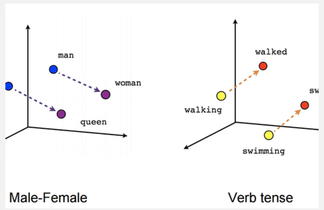

Figure 1: Visual Example of Word Vectors.
Taken from: https://www.synthesisproject.org/resources

3. Data Analysis

An algorithm called agglomerative clustering was selected to compare the sum of vectors for each tweet and pair groups of similar data together, to later explore such clusters' similarities to assign them a label as a category.

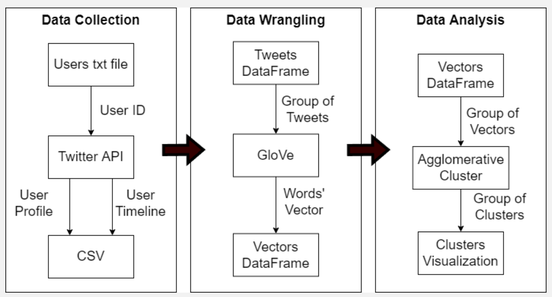

Figure 2: The Data Processing Pipeline.

Methods & Analysis

Words' Vector and GloVe

A 50-dimensional embedding of words was used to obtain the word's corresponding vector.
A word has a vector, which is a direction and value. Since GloVe has pre-trained vectors, each word has a pre-existing position, which on a 2-dimensional graph, these words would have an (x, y) coordinate. But since they are 50-dimensional embeddings, each word has fifty numbers describing their position instead of two. See figure 3. By summing the vector of each word in a tweet, and then summing the tweet's vector for 50 tweets of each user, which will be called the total vector, the overall significance of what the user engages with is captured. See figure 4 and 5 above for total vectors.

```
embeddings.keys()
embeddings['the']
✓ 0.1s
array([ 2.5320e-01, -1.4884e-02,  5.9371e-01,  1.5902e-01,  1.2754e-01,
        2.2428e-01,  8.9421e-01,  3.6396e-01, -3.1339e-01, -5.1857e-01,
        2.9637e-01, -4.1098e-02, -6.4555e+00,  3.2260e-01,  3.7288e-01,
       -6.1690e-01,  4.6744e-01,  5.0600e-01,  3.1950e-02,  1.0155e-01,
       -1.9615e-01,  1.3364e-01, -2.7140e-01, -4.1728e-01,  7.7940e-03,
        1.3573e-01, -7.2992e-02,  2.5208e-01,  5.1148e-01,  1.5120e-01,
        8.4398e-02, -2.4791e-01, -1.5913e-01,  1.5005e-01,  7.7243e-01,
        3.6632e-01, -9.8310e-02, -6.4317e-02, -7.1983e-04, -1.5231e-01,
       -1.4604e+00, -3.1696e-01, -4.1762e-01,  7.3363e-02,  3.2043e-01,
        3.4324e-01,  1.0895e-02, -2.8932e-01,  4.5493e-01,  1.8659e-01])
```

Figure 3: vector for the word "the"

```
df5.iloc[0]['total vector']

array([ 4.19994137e+01,  3.31788917e+01, -2.40983482e+00,  3.03023120e+00,
       -1.50037100e+00,  1.28843141e+01,  7.34155194e+01, -2.85221800e-01,
        2.63482286e+00, -4.82924848e+01, -1.26752476e+01, -2.54231288e+01,
       -5.68301911e+02,  2.55302700e+01,  2.51624730e+01,  1.76836334e+01,
       -9.60933800e+00, -1.07489195e+01, -1.11543877e+01, -2.52888100e-01,
       -1.40169660e+00, -1.77931690e+00, -6.01793760e+00,  6.62150230e+00,
        1.50031881e+01,  5.76499016e+01, -4.83655800e+00,  1.44331666e+01,
        2.32941024e+01,  1.12115497e+01,  1.87163465e+01, -1.46073069e+01,
        3.62925330e+00, -1.05413134e+01,  6.28762287e+01, -5.64023880e+00,
       -2.80680509e+01,  3.17998816e+01, -9.77973200e-01, -4.49303646e+01,
       -6.17706179e+01, -5.41413410e+00,  2.73893369e+01,  1.98153170e+00,
        8.42352980e+00, -1.63069185e+01,  3.12942777e+01,  4.81038500e-01,
        4.83942860e+00,  2.20368580e+00])
```

Figure 4: Total vector for user 0

tweet	clean	total vector
@BridgetRebecca4 @KayeSteinsapir My 36 year ol...	bridgetrebecca kayesteinsapir my year brother ...	[41.99941369999999, 33.1788917, -2.40983482000...

Figure 5: user 0's tweets, cleaned tweets, and total vector

Agglomerative Clustering

Agglomerative clustering is an algorithm used to group data. By allowing the computer to group similar total vectors, the groups can be later studied and analyzed for similarities, and then categorized. For example, below is a graph of three different clusters. Each cluster represents a group of tweets that have similar overall vectors. Referring to figure 6, the red points could be twitter accounts with total vectors closer to politics, while the blue points could be twitter accounts with total vectors closer to environment, and the green points could be total vectors closer to economics.

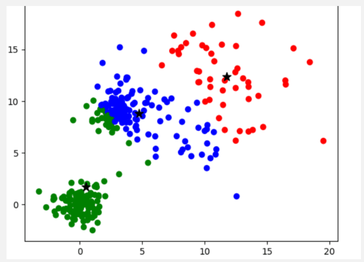

Figure 6: an example of agglomerative clustering
Taken from: https://dev.to/piyushbagani15/k-means-clustering-25hf

```
for i in range(8):
    print(f"Cluster {i}: " + str(df[df['results'] == i]['username'].values[:5]))

Cluster 0: ['11alynn11' '11bluebubbles' '1225rail' '1bomberfan' '1cdnguy']
Cluster 1: ['1garsvert' '29robidas' '50buts' 'ab_herboriste' 'adeline_weisse']
Cluster 2: ['1_tmf_' '350_pdx' '350canada' '3blmedia' '4fatherofmabbas']
Cluster 3: ['100possibleca' '10adge' '141_dee' '17einstein' '17kurri']
Cluster 4: ['11damdave' '1loriking' '222_gloria' '312shelly3' '350ma_metrowest']
Cluster 5: ['1denmadrid' '_solpost_' 'aantoniorg' 'alobarbogarde' 'alphafuego23']
Cluster 6: ['01_sonndeg' '04dolley' '0utputt' '0xrawzzay' '17mintz']
Cluster 7: ['00hazel' '01234566789a' '0xskye' '1101110112' '12wandler']
```

Figure 7: 5 Twitter users from each cluster

Analysis

The 30 000 Twitter users' latest 50 tweets were extracted, and then cleaned with the program NLTK so only the nouns were left. Only the nouns were analyzed since words like "I", "had", or "is" do not have any significance to the users' overall engagement. Then, the 50 tweets of each user were processed in GloVe for a total vector. After inputting all the users' total vectors in the clustering algorithm, and inputting 8 as the desired number of groups, the computer created a graph with 8 clusters, see figure 8. Each cluster is a group of similar total vectors. To find the optimal number of clusters for a data set, the elbow method is used, see figure 10.

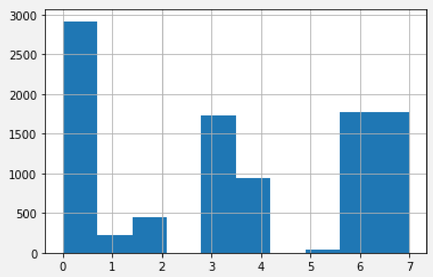

Figure 8: Graph of the 8 clusters

```
df['results'].value_counts()

0    2916
7    1775
6    1768
3    1726
4     943
2     441
1     228
5      46
Name: results, dtype: int64
```

Figure 9: The number of Twitter users in each cluster

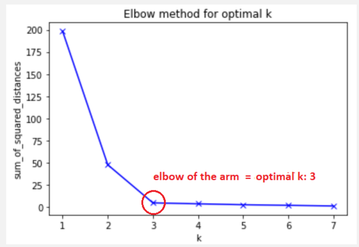

Figure 10: the sum of squared distances as a function of the number of clusters
Taken from: https://stackoverflow.com/questions/62443970/finding-the-optimal-number-of-clusters-using-the-elbow-method-and-k-means-clust

Results

Face Validity

After examining some user accounts from each cluster, some similarities were found. In some cases, these similarities were obvious, but for the most part, there was a lot of overlap. Below are some generalizations made after examining 5 users from each cluster.

Cluster 0: anti-Trudeau, right wing
Cluster 1: French text
Cluster 2: climate change activists
Cluster 3: left wing
Cluster 4: economy activists
Cluster 5: Spanish text
Cluster 6: miscellaneous
Cluster 7: not clear

The only two clusters that were strongly consistent were cluster 0, cluster 1, and cluster 5. The other clusters were moderately consistent, while cluster 7 did not have a clear theme.

When analyzing the tweets in each cluster, tweets were grouped together based on the object of conversation, but not the sentiment. For example, a user who tweeted: "I hate Trudeau", and another who tweeted: "Trudeau is great", were placed in the same cluster since the noun of both sentences is "Trudeau." This made it hard to find consistent similarities in each cluster.

Additionally, some of the tweets were image-based, to which the user only tweeted emojis or basic sentiments as text. This was not properly analyzed by GloVe since, as stated above, GloVe cannot capture sentiment. This too made the data hard to successfully cluster.

Conclusions

By using the GloVe program and the clustering algorithm, the 30 000 Twitter users were categorized into 8 groups. By having a model that can categorize users based on the meaning of words, the perspectives of each group can be studied for better policy making. This fits into the bigger project as it allows for demographic trends in persuasion and manipulation tactics to be studied.

Next Steps

The 8 clusters were moderately consistent, however, some clusters had a lot of overlap. To mitigate this, sentiment analysis should be used to cluster users not only on the topic discussed, but also on their position towards it. Additionally, more investigation is required to identify the ideological affinities present in each cluster. Lastly, more calculations on the optimal number of clusters is required.

Acknowledgements

Thank you Dr. Lefsrud, Dr. Stroulia, Mr. Roberts, and Mr. Gutierrez for helping me with this project, and thank you AI4Society and Dr. Lefsrud for sponsoring this project! I would also like to thank everyone else who helped me along the way, and finally, I would like to thank the WISEST team.