Social Media Mining to Study Social User Group by Visualizing Tweet Clusters using Word2Vec, PCA and K-Means

Md Morshed Jaman Adnan[a], Matthias L. Hemmje[b] and Michael A. Kaufmann[c]

[a] *Hochschule Luzern – Informatik, Campus Zug-Rotkreuz, Suurstoffi 1, 6343 Rotkreuz, Switzerland*
[b] *FernUniversität in Hagen, 1 F09, Informatikzentrum, Universitätsstr. 1, 58097 Hagen, Germany*
[c] *Hochschule Luzern – Informatik, Campus Zug-Rotkreuz, Suurstoffi 1, 6343 Rotkreuz, Switzerland*

Abstract

In social media mining, unsupervised machine learning has not been studied extensively. Based on a specific use case in political polarization detection, we contribute an experimental method to find clusters in social media data and to explain the semantics of cluster membership using frequent word mentions. 100 dimensional word2vec word embedding vectors are generated from the raw text data. Based on this, three clustering approaches were tested: K-means, K-Means with Principal Components Analysis (PCA), and Deep Embedded Clustering (DEC). For K-Means, the optimal number of clusters was estimated visually. The performance of the clustering approaches was compared using the Silhouette score. Using PCA to reduce the dimensionality from 100 to only 2 most significant principal components had a significant performance impact, and combined with K-means, provided the best results. To explain the clusters visually, word counts of the most frequent words were plotted for three clusters. It became evident that the most frequent words per cluster explained the semantics of cluster membership in a consistent way. We conclude the following: our method demonstrates it is possible to find and explain meaningful clusters of users in social media data. Dimensionality reduction of text based on Word2Vec and PCA led to semantically coherent, clearly distinct clusters. Based on our understanding of the underlying use case, the clusters generated by the proposed method clearly reflect social phenomena that can be verified by a qualitative interpretation of the cluster visualizations. Thus, the proposed method can provide a basis for studying social behavior using social media data.

Keywords 1

Social Media Mining, Explainable Clustering, Cluster Visualization, Interpretability, Computational Social Science

1. Introduction

Social media networks serve as a platform for sharing multimedia objects and establish interactions between users in the form of various activities. For instance, Twitter is a famous micro-blogging platform where Users can post up to 140-character long messages called Tweets [1]. These are publicly available, providing readily useable social media data. This data can provide useful information in various domains and can possibly be used for extracting meaningful features about social phenomena.

For instance, social media platforms such as Twitter, Facebook have the potential to amplify peer effects and social division in political behavior. According to Barbera [2] people depend on their personal networks to gather political identities and make their voting decisions. The hypothesis of this research is that social behavior such as this can be analyzed in open-source data of social media networks. Pew Internet and American Life Project reported that every six out of ten Internet users and

BIRDS 2021: Bridging the Gap between Information Science, Information Retrieval and Data Science, March 19, 2021. Online Event.
EMAIL: mdmorshedjaman.adnan@stud.hslu.ch (M. Adnan); Matthias.Hemmje@FernUni-Hagen.de (M. Hemmje); m.kaufmann@hslu.ch (M. Kaufmann)

nearly 44% of American adults went online to get news or information about politics [3]. The presence of political polarization in social media, blogs and other web 2.0 platforms may be explained by a well-known phenomenon in sociology called homophily [4] where people in social media networks tend to group around with similar backgrounds and interests, including shared political views. During the 2011 Canadian Federal Election, there was a significant discussion about the influence of the social media [5]. According to Bennet, [6] social media plays an important role in shaping political discourse in the U.S. and around the world. The specific use case of this study is to test if this behavior can be recognized directly in social media data.

It is technically possible to apply data mining algorithms to social media datasets and find patterns and relationships. The hypothesis of this research is that these algorithms can realistically reflect social interactions. For example, clustering can be used to extract groups of users from unlabeled data. In this study, unsupervised clustering technique was studied to test if it is possible to group users who share similar semantically thoughts in their social media communications. For this, three clustering algorithms are compared on tweets collected on the U.S. 2016 election day. The clustering methods take generated features of social media data as input, and generate and clusters with similar attributes. Accordingly, the clusters can be analyzed to verify if it is possible to observe different political motivations in the cluster analysis. The objective of this research was to check if it possible to find politically orientated communities of 2016 U.S. presidential election. The research questions of this research were the following:

- RQ1: How can the result of the social media data clustering be presented, visualized and / or communicated in a way that supports empirical conclusions about social phenomena?
- RQ2: Which data structure is optimal for human- oriented data analysis to generate explainable and trustworthy insights?
- RQ3: Which clustering analysis approaches are suited for knowledge generation, explainable insights and actual conclusions (instead of predictions or correlations)?
- RQ4: How can we deduce some form of support for the proposition that the given data and its analysis support any conclusion about actual real-world phenomena?

This paper is organized into five chapters. This introduction provides context and motivation for the described research. Section 2 explains the background and the current state of the art. Section 3 gives a detailed overview of the methodological approach where it is explained the data preparation and analysis. In Section 4, the results of the cluster analysis are carried out by quantitative cluster performance evaluation. Finally, the conclusion of this research, backed by the generated data, is described in Section 5.

2. State of the Art

2.1. Social Media Mining

Social media mining (SMM) refers to the analysis of massive social media data using machine learning tools and techniques. The interdisciplinary field combines aspects of various domains such as computer science, data mining, machine learning, information retrieval, and social network analysis. The objective of SMM is to measure, model extract meaningful patterns from large-scale social media data. This data generated from different social media platforms are unstructured and noisy [7] due to the use of various short forms, special characters, limitations of the use of number of characters, symbols. Because of the casual nature of social media, its users often use colloquial forms of languages [8]. Furthermore, social media content is generated by its users in unstructured format, which makes it more difficult for machine learning algorithms to find important insights. Therefore, data preprocessing is central to find important patterns from the social media data.

Twitter is a social and microblogging platform where users can post up to 140-character long messages, or Tweets. Besides broadcasting tweets to an audience of followers, Twitter users can interact with one another in two primary public ways: retweets and mention. Retweets act as a form of endorsement, allowing individuals to rebroadcast content generated by other users. In this study, Twitter data collected on the 2016 election day was used to find if a data clustering reflects actual social groups.

2.2. Clustering Techniques

The issue of identifying communities in large social networks is an important contemporary challenge of computational social science. Clustering social media data can make this task easier. Clustering is a machine learning technique that divides the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. In the area of data science, clustering techniques are used to gain insights from the data by finding equivalence classes in the data automatically. There exist various types of clustering algorithm such as K-Means, Mean-Shift clustering, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), Agglomerative Hierarchical clustering and so on. Performance of clustering can be defined by a score, such as the Silhouette metric, which measures how well the data points are classified by the clusters. Thus, different clustering techniques can be compared quantitatively.

Since K-Means clustering algorithm is widely used in data mining, this technique was chosen in this research to examine different approaches of clustering algorithms. Also, a neural network based approach, deep embedded clustering, was compared.

2.3. K-Means Clustering

K-means is an unsupervised machine-learning algorithm well known as a clustering technique [9] commonly used to automatically partition a dataset into k groups. It is a distance-based clustering algorithm that divides data into a number of clusters using numerical attributes. However, it is well known that the K- Means algorithm suffers from the serious drawback that its performance heavily depends on the initial starting conditions and the number of classes [10]. The working principle of K-Means clustering is explained in the following.

- Input: K (the number of cluster) D (a set of lift ratios)
- Output: a set of K clusters
- Initialization: Arbitrary choose K objects from D as the initial clusters
- Repeat until no change:
    1. (Re) assign each object to the cluster to which the object is the most similar based on the mean value of objects in the cluster
    2. Update the cluster means, i.e, calculate the mean value of the objects for each cluster.

2.4. K-Means with PCA Dimensionality Reduction

The distance metric of K- Means clustering algorithm is limited to the original data space and tends to become ineffective when the input dimensionality is high. To resolve this issue, the dimension of the vector space can be reduced to a lower dimension with a dimensionality reduction algorithm. Then, K-Means can be applied to the reduced dimension later on.

High- dimensional data often suffer from a term called “curse of dimensionality” [11] The number of samples needed to estimate and arbitrary function with a given level of accuracy grows exponentially with respect to the number of input variables of the function. In higher dimensional space, data points located to be far away from each other make the clustering harder and become vulnerable to over-fitting [11]. For example, a quartering of numeric dimensions leads to 4 regions in 1-D, 16 regions in 2-D and 64 regions in 3-D, and the regions become sparser, i.e., show less points, with increasing number of dimensions. Therefore, in high-dimensional space, the given data fills less and less of the data space. To make up for sparsity in high dimensions, a large number of data points would be needed. A different approach to the curse of dimensionality is to use a dimensionality reduction technique, for example, Principal Component Analysis (PCA).

PCA is a widely used statistical technique that can be applied for unsupervised dimension reduction [12]. In other words, PCA [13] is a data modeling method that produces a ranked list of new dimensions, called principal components. Taking the top n principal components for further

processing is often used to reduce the dimension into a lower dimensional space where coherent clustering can be detected more clearly. Principle components represent dimensional vectors of the data that explain a maximal amount of variance, that is to say, the lines that capture most information of the data using least squares. First, PCA is a reconstruction of the latent space from the original data space, and it does not lead to a loss of information in the first place. Principle Components are constructed in a ranked manner that the first principal component accounts for the largest possible variance in the dataset, the second component for the second largest, and so on. Selecting the most important components in this list leads to a dimensionality reduction with minimal information loss. Although, there exists various techniques for dimensionality reduction, the PCA is by far the most popular (unsupervised) technique [14].

2.5. Deep Embedded Clustering (DEC)

Deep clustering automates a combination of feature extraction, dimensionality reduction and clustering. A deep embedded clustering approach has been described by [15]. The DEC method can be explained in two phases that include parameter initialization with deep auto-encoder and parameter (clustering) optimization. The idea of DEC is a parameterized non-linear mapping from the data space X (original data space) to a lower-dimensional feature space Z, where the clustering objective is optimized. Unlike previous work, which operates on the data space or a shallow linear embedded space, the authors [15] use stochastic gradient descent (SGD) via back propagation on a clustering objective to learn the mapping, which is parameterized by a deep neural network. The DEC neural network architecture has an encoder layer, decoder layer and a clustering layer. The encoder takes the higher dimensional data as input and suppresses them into a lower dimensional latent space Z. This latent space Z is now added to the clustering layer, which is termed DEC by Xie [15]. The proposed DEC is performed in two phases, soft labeling and joint refinement. These two phases involved in data preparation and training the data for the proposed clustering model. Therefore, these two phases are explained in the data preparation and analysis chapter.

3. Method

3.1. Conceptual Approach

This subsection summarizes the method of data analysis applied in this research, and it works as a bridge between the state of the art and the data preparation and analysis. As input, the raw text of a collection of Tweets was taken. The raw data was preprocessed to create a high-quality text corpus. An exploratory analysis of the text corpus was performed, yielding keywords of the corpus. Using a word2vec model, words were transformed into vectors. Tweets were transformed into average feature vectors (centroids). Optionally, the number of dimensions was further reduced two 2 using PCA. A clustering was performed using K-Means with and without PCA and with DEC. The Silhouette score for the resulting clustering were measured and compared. For the best performing clustering technique, the clusters were visualized by plotting word mention frequencies for keywords found in the exploratory analysis. The resulting visualizations of clusters were analyzed and interpreted to infer relationships between the components of specific clusters explained with figures.

3.2. Data Collection

The Twitter data used in this research is collected from Kaggle, which contains filtered tweets of the 2016 U.S. Election Day in csv format. The dataset consists of four hundred thousand tweets and comes with several fields including created at, language, original tweets, username, user location, user id, user follower count, user friends, user profile, user time zone, favorite count, user followers, user status count, user favorite, retweeted, retweet count. The original tweet text column contains the tweets posted by users in Twitter. This dataset most likely contains political discussion on the election polls as they are generated on the day of U.S. election in 2016. The use case for selecting this dataset

is to test if it is possible to observe user groups who write about similar topics using clustering techniques directly on digital data, and in this case, communities with different political language.

3.3. Preprocessing

Text pre-processing is an important aspect of natural language processing and refers to the practice of cleaning and preparing text data for further analysis. Especially, raw social media data comes with various unnecessary noises such as html contents, hyperlinks, special characters, unrecognized symbols etc. Noisy data refers to the existence of meaningless or corrupted data in the datasets. Noise can have significant impact on the overall performance of a machine-learning model. For instance, the noise can result in errors in the prediction of machine learning algorithms and can impact their performance in terms of accuracy, size of the model and the time taken to build the model [16]. There exists large amount of noise in the dataset, which include unknown symbols, special characters and words written in short forms (ppl, im, dem) etc. Such contents barely contribute to the feature extraction and thus, it was filtered using a pre- processing function.

Accordingly, a pre-processing function was written in python 3.7 to prepare the dataset for feature extraction in the next step. For the simplicity of analysis, only the English tweets were considered. The pre-processing function removed punctuation, regular expression, stopwords, https, single letter, and special characters from the original tweets. In addition to that, stemming and lemmatization techniques were applied to reform the words into their root form. The implementation of the pre-processing function reduced the tweets to the length of 134960 that accumulates one third of the whole corpus.

3.4. Exploratory Data Analysis

One way to get valuable insights from any data is to visualize them. However, the visualization is a critical part of any data analysis and makes it easy for humans to understand the significance of the data. In the present context, it is important to check if the collected dataset is relevant for the desired analysis. Accordingly, the 10 most frequent words in the dataset were calculated and found that the words election2016, congress, democrat, republican, trump, vote etc. appeared most often as depicted in the Figure 1. Therefore, it can be stated that the collected twitter dataset contains words of different politically motivated users and it is possible, accordingly meaningful patterns later in the cluster analysis section.

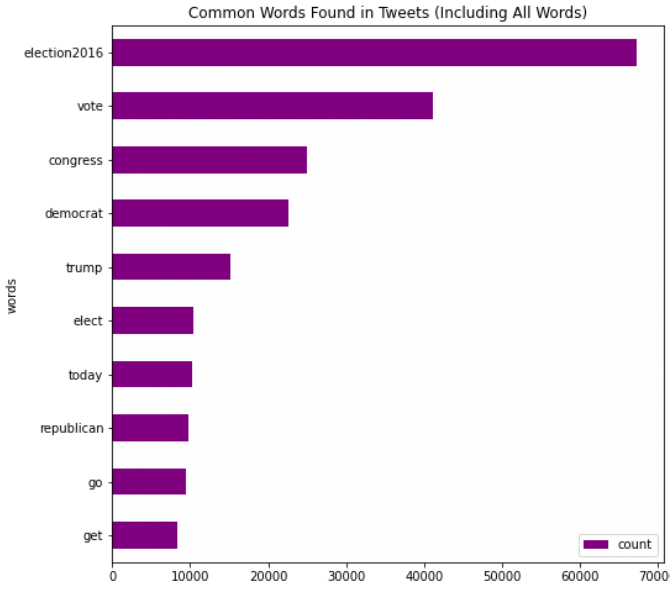

Figure 1: Words with high frequency in the Tweets

These frequent words can be recognized as keywords and it is possible to investigate if these frequent words have different distributions in any subsequent clusters later in the analysis part.

3.5. Feature Extraction

Machine-learning algorithms cannot directly work with the text; they require some transformation of texts to vector of numbers. Word2vec is a two-layer neural network that process text by vectorizing the words developed by Miklovo [17]. The model takes text corpus as input and produces a set of vectors into several hundred of dimensions as output.

Using the implementation of Word2vec, the pre-processed dataset was feed into the model and a 100-dimensional vector was generated each word. The words sharing similar contexts also share a similar meaning and consequently a similar vector representation from the model. The word2vec model was trained on the text corpus and produced centroid vectors for tweets by calculating average feature vectors, which produced a shape of (134960, 100) equal to the length of pre-processed dataset.

3.6. Estimating the Optimal Number of Clusters

Unlike Self- Organizing Maps and DBSCAN, both K- Means and DEC clustering algorithms cannot automatically formulate the optimal number of clusters. By convention K- Means and DEC require the number of K cluster as input. There exist various measures such as Silhouette score, Elbow methods etc. to decide the number of clusters to be formulated from the data. For instance, Silhouette is a score between -1 to +1, used to assess the validity of clustering [18] performance. A high Silhouette value indicates that an object is well matched to its own cluster and distant from the neighboring clusters [19]. In this paper, the Elbow method of Scikit Learn is used to determine the optimum number of clusters, using Distortion score as a performance metric, which computes the sum of squared distances from all data points to their assigned centers.

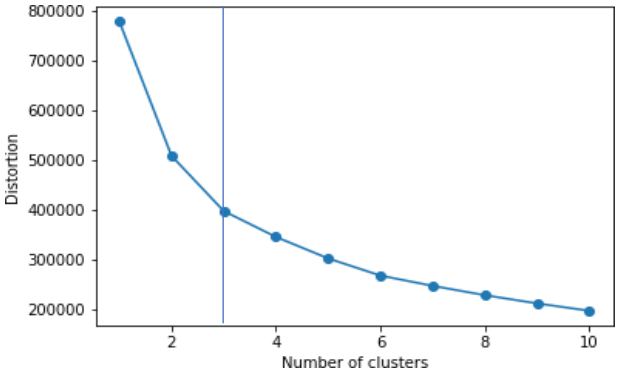

Figure 2: Visualization of the Elbow method for finding the optimal number of clusters

The elbow method in Figure 2 shows that, the elbow or knee point, defined as the point at which the curve angel becomes less than 45°, is located at k=3. Therefore, k=3 is considered the optimal number of clusters on this dataset.

3.7. K-Means Clustering on the Tweet Centroid Word Vector

A K-Means clustering algorithm was applied to the 100-D data space resulting directly from the feature extraction method described in Section 3.5, using the number of clusters described in Section 3.6. From this, 3 clusters were derived. The Silhouette score was calculated on the K- Means clusters based on Euclidean distance, as described in Section 4.1. Also, a two-dimensional projection of the clustering and the resulting cluster centers was visualized, as shown in Section 4.2.

3.8. K-Means With PCA

Inspired by the idea of [20] using PCA to project the data into lower dimensional sub- space and K-Means clustering applied to the sub- space, a similar clustering approach has been applied in this analysis. The original feature space was high dimensional (100-D). To reduce the dimensionality of the data, PCA technique was used to project the input data in 2 dimensional spaces by taking into account the two most principal components. In this model, Sklearn's PCA implementation was applied, which uses the LAPACK implementation of the full Singular Value Decomposition (SVD) or a randomized truncated SVD depending on the shape of the input data and the number of components to extract. In this implementation the number of components was 2.

While examining PCA on the higher dimensional dataset, the results of various n components values were observed. The original shape (1349600, 100) of the data was transformed to the reduced shape of (1349600, 2). K-Means clustering was now applied to the lower dimensional sub-space to formulate clusters. It was found that, n=2 principal components provide best Silhouette score in the cluster analysis. The original shape (134960, 100) of the input data was transformed to the reduced shape of (134960, 2). K-Means clustering was now applied to the lower dimensional sub-space to formulate clusters.

3.9. DEC Deep Embedded Clustering

Creating and training Auto-encoder: As an alternative to PCA, deep learning dimensionality reduction was also performed in the dataset to test for the difference. The vector representation of each word in the dataset was calculated to 100 dimensions, as described above. It was then possible to apply deep learning dimensionality reduction technique to compress the core information of the higher dimensional dataset. The deep data compression model resulted in a stack of encoder layers and took the higher dimensional data as input and compressed them into lower dimensional latent space. Neural networks are trained using stochastic gradient descent and require appropriate loss function depending on the configuration of the model. Loss functions are used to calculate the model error in general. The goal of training the stacked auto-encoder model that is explained in figure 5 is to minimize the loss of the training error. The loss describes the objective that the auto- encoder tries to reach. The model was configured with sigmoid activation function and the mean squared error loss function. As depicted in figure 5, the model encoder layer was increased by two more layers with the dimensions of 3000 and 5000 compared the current state of the art model [15]. The reason behind the increased number of layers in the model was to investigate whether deep learning model achieves better performance with larger networks stated by [21]. The stacked encoder model was trained with 128 batch-size and 50 epochs. The training of the auto-encoder model was carried out in Google Colab pro version and the loss was reduced to 0.2099 from 0.0137 after 50 epochs. However, the hyperparameters of the neural network model were trained with default settings except the increased dimensions in the encoded layers of the input data space. The network could produce better result with tuned hyperparameters. For instance, the model could have been trained with more epochs and with different batch size. However, the training of the encoder model takes significantly enormous time with such large network in Google Colab pro.

DEC Soft Labeling: One of the key components of the proposed DEC model is the soft labeling. The idea of soft labeling is assigning an estimated class to each of the data samples in such a way that it can be refined iteratively. A clustering layer was added to the auto-encoder output.

Joint Refining DEC: Optimizing DEC is challenging. The goal is to simultaneously solve for cluster assignment and the underlying feature representation. However, unlike in supervised learning, the deep network cannot be trained with labeled data. Instead, an iterative refining cluster method is proposed with an auxiliary target distribution derived from the current soft cluster assignment. This model used the pre-trained auto- encoder and a K-Means model to define a new model that takes the pre-processed twitter data as input and produces output both the predicted clustering classes and decoded input data records. The model summary is depicted in Figure 3.

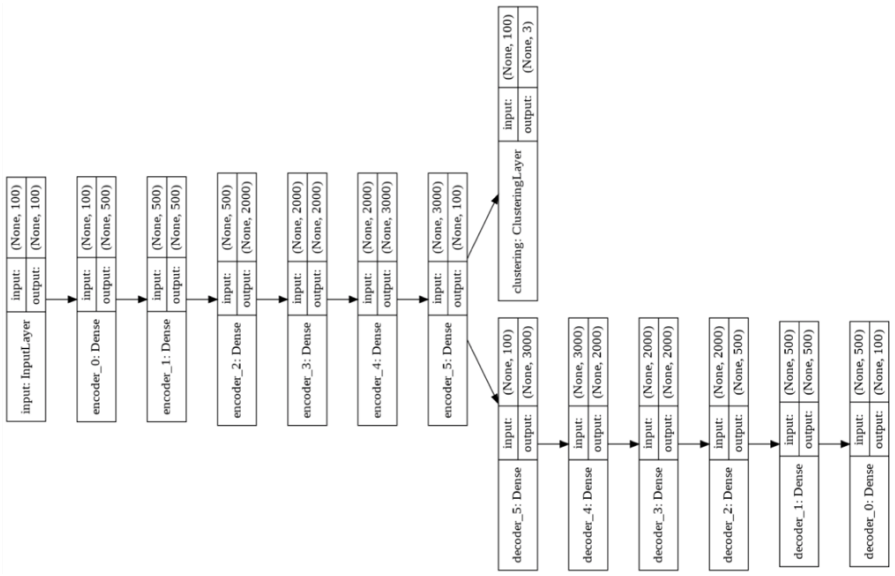

Figure 3: Joint Refining DEC model summary

4. Results

4.1. Quantitative Cluster Performance Evaluation

For the quantitative evaluation of clustering performance, the Silhouette score was measured. The Silhouette score of the K-Means clustering on the 100-D word-vector centroids was 0.0641, which indicated that the data points, on average, were not well matched to their own clusters and too close to the neighboring clusters.

It was found that a PCA-based 2-D data space achieved a much better Silhouette. It was observed that the K-Means with 2-D PCA outperformed K-Means clustering of the 100-D space significantly in terms of the Silhouette score. PCA based K-Means achieved a Silhouette score of 0.361 whereas the K- Means scored 0.064. However, a Silhouette score of 0.75 is usually required for a cluster to be considered good [15].

It was assumed that the DEC model would improve the clustering process as well as the feature representation. However, it was observed that the DEC refined model achieved a Silhouette score of 0.0415661, which is the lower than the K-Means clustering shown in table 1. Moreover, it was surprisingly noticed that DEC Soft labeling deep clustering achieved a Silhouette score of 0.495. In Table 1, the resulting Silhouette socres are summarized.

Table 1: Clustering methods with resulting Silhouette score

Clustering Approaches	Silhouette Scores
K-Means 100-D	0.0641
K-Means 2-D (PCA)	0.3619
DEC Soft Labeling	0.495
Joint Refining DEC	0.0415

4.2. Cluster Visualization and Qualitative Evaluation

To compare the resulting clustering qualitatively, the feature space was projected to a two-dimensional space, the data points were plotted, and their cluster membership encoded in the dot color. For K-Means, the top two PCA-components were used; for DEC, the first two dimensions generated by the auto-encoder were used to scatter-plot the data points in 2-D. The first method, K-Means 100-D resulted in the graphic shown in Figure 4a.). The cluster centers were fused together as

shown in the graphic The vector space, in this case, was a high-dimensional 100-D representation of the data set, and K-Means became less effective at distinguishing between data points in the high dimensional space. When the data space was reduced to two dimensions using PCA, the cluster visualization resulted in the graphic shown in Figure 4b.). The clusters centers were well separated in the PCA based K-Means clustering. DEC Soft labeling created the following cluster centers as shown in Figure 4c.) The DEC Soft labeling showed a dominance of one cluster (in purple). The very imbalanced clustering was not usable; therefore, a new approach was tried, called Joint Refining DEC. The Joint Refining DEC approach resulted in the graphic shown in Figure 4d.) The cluster colors were randomly distributed, and it was difficult to identify the clusters by looking at the graphic. Like K-Means clustering on the 100-D space, the cluster centers were also fused in Joint Refining DEC.

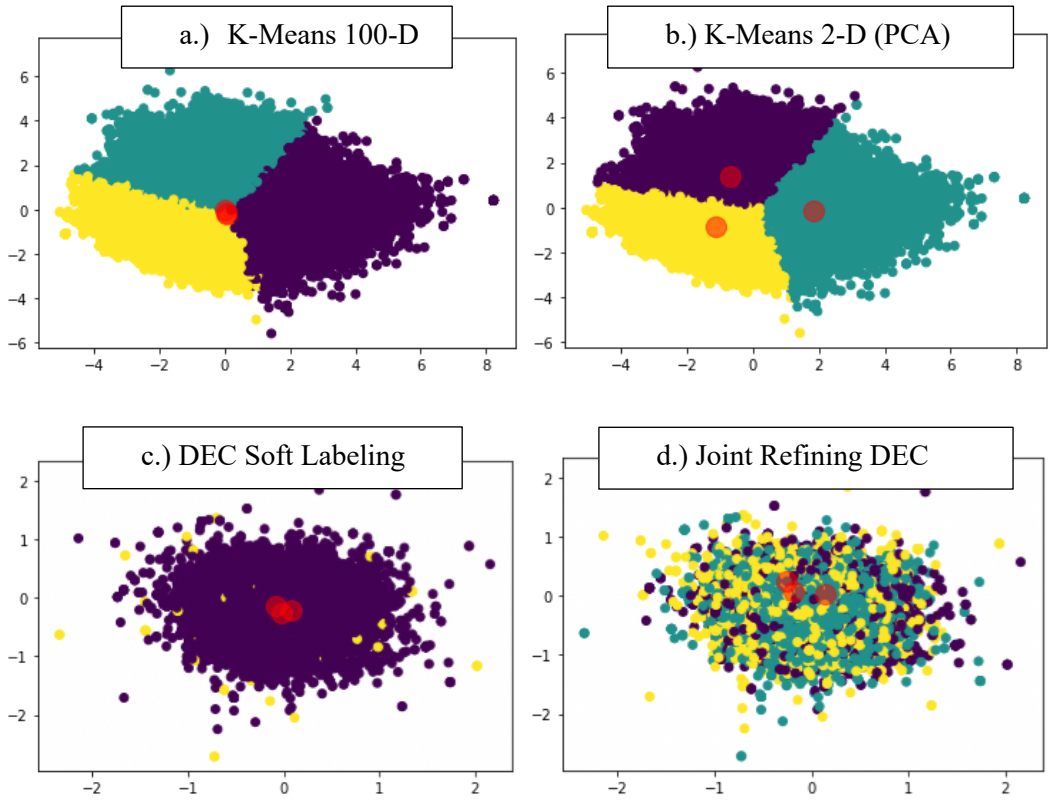

Figure 4: Cluster visualization projected on a lower-dimensional feature space: a.) K-Means clustering on 100-D word vector centroids, b.) K-Means on the two most important principal components (PCA), c.) DEC with soft labeling, and d.) Joint Refining DEC.

Although the first part of DEC, DEC Soft Labeling showed the best Silhouette in Table 1, the plot in Figure 4c.) revealed qualitatively that this method was not usable in practice because it was highly unbalanced and resulted, de facto, in only one dominant cluster, and some outliers. Therefore, the quantitatively second-best method, K-Means 2-D (PCA), was chosen for cluster explanation and visualization, which is described in the next section. However, the final stage of DEC method which is DEC Joint Refining shows significantly lower score Silhouette score than the K-Means on 100-D word vector. Therefore, the quantitatively the best method, K-Means 2-D (PCA), was chosen for cluster explanation and visualization, which is described in the next section.

4.3. Visualizing cluster semantics with word frequency distributions

The clusters results of the PCA + K-Means method described above was analyzed based on keywords found in the exploratory data analysis. Words with high frequencies were examined in the dataset and it was possible to investigate if these high frequent words have different distributions give

the clusters. Unlike [3] where the clusters were analyzed with Hashtags, in this work clustering was performed on the keywords without Hashtags.

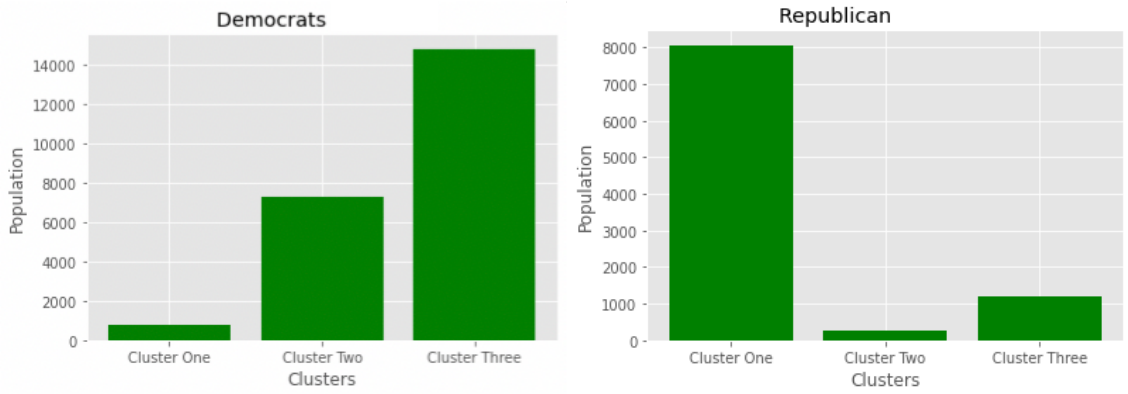

Figure 5: Number of mentions of the word “democrats” and “republicans” in the clusters

Figure 5 represents the clusters of the number of mentions of the keywords “democrats” and “republicans” in the twitter dataset. The number of the mentions of the keyword democrats (22842) in the cluster is higher than the number of mentions of the keyword “republican” (9506). The number of mentions of the keyword democrats is significantly lower than the number of the mentions of the keyword republican in cluster one. However, the number of mentions of the keywords democrats and republican shows opposite patterns than the cluster one.

Figure 6: Number of mentions of the word “Trump” and “Hillary” in cluster

The number of mentions of the keywords “Trump” and “Hillary” are depicted in Figure 6. The cluster results shows that the number of mentions of the keyword “Trump” (20612) appeared three times higher than the “Hillary” (9645) in a cluster.

Figure 7: Number of mentions of two polarizing keywords “Lockherup” and “MAGA” in cluster

The word frequency distributions of the keywords “LOCKHERUP” and “maga” are illustrated in Figure 7. Overall, the keyword “maga” appeared more than the keyword “LOCKHERUP” in the clusters. Cluster three held the highest relative number of both the keywords. Cluster two showed less mentions of these polarizing keywords, and cluster one, one on the other hand, contained only a few of these keywords.

5. Conclusions

Regarding RQ1, about visualizing the clustering result, the following answer is found. The results of the social media data clustering are visualized using word frequency distributions over the clusters for frequent words in the corpus. The number of mentions of the various frequent keywords such as democrats, republican, Trump, Hillary etc. were visualized in each cluster. Qualitatively, by looking at Figures 9-11, it seems to be possible to assume that for example Cluster 3 seems to be a cluster of right-wing users, whereas Cluster 2 seems to be the left-wing cluster. Thus, we find some evidence that that is possible to infer cluster semantics from these word frequency distributions over the clusters.

With regard to RQ2, about the database format, this research proposes the corresponding answer. The original twitter dataset is collected in csv format where python data frame is primarily used to visualize the data. The original data was transformed into 100-D centroid word vectors per Tweet, generated from word embedding model. Furthermore, the best results were achieved by reducing the dimensionality to two principal components using PCA in a second step.

With reference to RQ3, about clustering techniques, four different clustering approaches were compared in this research. The results of each approach were recorded. The qualitatively (As assessed by a 2-D plot of the clustering) and quantitatively (using Silhouette score) optimal results were achieved using a K-Means clustering together with 2-D PCA and k=3, as determined by the elbow method using a Distortion score.

With regard to RQ4 about reality correspondence, users use common languages in social media and these languages have semantic meanings. Our propose cluster visualization can reveal these meanings by showing the word frequency distributions over clusters. However, without some sort of ground truth, it is impossible to prove that the clusters identified are semantically optimal, or indeed whether the relative performance of the algorithms is heading toward any notion of optimality. Also, any clusters identified can behave somewhat like a Rorschach ink blot: we see the patterns in them that we expect to see. Also, the choice of keywords to visualize is an important decision. We chose the most frequent keywords, but other techniques such as tf*idf could be investigated. As such, The interpretation of the visual results is qualitative and subjective. Thus, a semantic differentiation between the clusters is possible using a qualitative, subjective interpretation of the objective quantitative word distribution measurement. In our first experiment, the distribution seems to correlate with similar semantic meaning. However, without some sort of ground truth, it is impossible to prove that the clusters identified are semantically optimal, or indeed whether the relative performance of the algorithms is heading toward any notion of optimality.

6. References

[1] M. D. Conover, J. Ratkiewicz, M. Francisco, B. Goncalves, A. Flammini, and F. Menczer, “Political Polarization on Twitter,” p. 8.

[2] P. Barbera, “How Social Media Reduces Mass Political Polarization. Evidence from Germany, Spain, and the U.S. – SMaPP: Social Media and Political Participation,” New York University, Working Paper, 2014. Accessed: Feb. 20, 2021. [Online]. Available: https://wp.nyu.edu/smapp/how-social-media-reduces-mass-political-polarization-evidence-from-germany-spain-and-the-u-s/.

[3] M. Conover, J. Ratkiewicz, M. Francisco, B. Gonçalves, F. Menczer, and A. Flammini, “Political Polarization on Twitter,” presented at the Fifth International AAAI Conference on Weblogs and Social Media., Jan. 2011.

[4] M. Mcpherson, L. Smith-Lovin, and J. Cook, “Birds of a Feather: Homophily in Social Networks,” *Annual Review of Sociology*, vol. 27, p. 415, Jan. 2001, doi: 10.3410/f.725356294.793504070.
[5] A. Gruzd and J. Roy, “Investigating Political Polarization on Twitter: A Canadian Perspective,” *Policy & Internet*, vol. 6, Mar. 2014, doi: 10.1002/1944-2866.POI354.
[6] L. Bennett, “Communicating Global Activism,” *Information*, vol. Communication&Society, pp. 143–168, Jan. 2003, doi: 10.1080/1369118032000093860a.
[7] C. Sengstock and M. Gertz, “Latent geographic feature extraction from social media,” in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL ’12*, Redondo Beach, California, 2012, p. 149, doi: 10.1145/2424321.2424342.
[8] N. Agarwal and Y. Yiliyasi, “Information quality challenges in social media,” *The 15th International Conference on Information Quality*, Jan. 2010.
[9] J. Macqueen, “Some methods for classification and analysis of multivariate observations,” in *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.
[10] D. Sisodia, L. Singh, and S. Sisodia, “Clustering Techniques: A Brief Survey of Different Clustering Algorithms,” *International Journal of Latest Trends in Engineering and Technology*, vol. 1, no. 3, p. 6, 2012.
[11] M. Verleysen and D. François, “The Curse of Dimensionality in Data Mining and Time Series Prediction,” Jun. 2005, vol. 3512, pp. 758–770, doi: 10.1007/11494669_93.
[12] “K-means clustering via principal component analysis | Proceedings of the twenty-first international conference on Machine learning.” https://dl.acm.org/doi/abs/10.1145/1015330.1015408?casa_token=HebDkUQJ5RkAAAAA%3AZYOLr1IxDDmeuOTq5Q0QCbSERAXCcoW9Sb0WJXkGtd5B75-dpl1sInXEKw09Z5LMt-jCDaPAicoQ (accessed Feb. 20, 2021).
[13] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York: Springer-Verlag, 2002.
[14] L. van der Maaten, E. Postma, and H. Herik, “Dimensionality Reduction: A Comparative Review,” *Journal of Machine Learning Research - JMLR*, vol. 10, Jan. 2007.
[15] J. Xie, R. Girshick, and A. Farhadi, “Unsupervised Deep Embedding for Clustering Analysis,” Nov. 2015.
[16] X. Zhu and X. Wu, “Class Noise vs. Attribute Noise: A Quantitative Study,” *Artif. Intell. Rev.*, vol. 22, pp. 177–210, Nov. 2004, doi: 10.1007/s10462-004-0751-8.
[17] T. Mikolov, K. Chen, G. s Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *Proceedings of Workshop at ICLR*, vol. 2013, Jan. 2013.
[18] T. Thinsungnoen, N. Kaoungku, P. Durongdumronchai, K. Kerdprasop, and N. Kerdprasop, “The Clustering Validity with Silhouette and Sum of Squared Errors,” Jan. 2015, pp. 44–51, doi: 10.12792/iciae2015.012.
[19] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987, doi: 10.1016/0377-0427(87)90125-7.
[20] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, “Spectral Relaxation for K-means Clustering,” *Adv. Neural Inf. Process. Syst.*, vol. 14, Apr. 2002.
[21] C. Szegedy *et al.*, “Going deeper with convolutions,” Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.