Department of Electrical and Computer Engineering

Part IV Research Project Final Report

September 2016

Grouping Similar Newspaper Articles

Chanjun Park, Ruoyi Cai (Partner)
Gill Dobbie (Supervisor)

Declaration of Originality

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

Name: Chanjun Park

Abstract— **There are many different methods that have been explored and researched that deal with grouping and clustering data. Some examples of such methods that have been identified are the K-Means clustering algorithm [1] or the Agglomerative clustering algorithm [2].**

Within this report we detail and present clusterr, a web application developed that has explored, tested and evaluated various methods to develop a way in which newspaper articles could be grouped and clustered by their contents. Additionally, we explore the current state of implementation and the possible applications that clusterr has, as well as possible extensions that are available for a service such as this.

I. Introduction

When considering the current progression of trends within the technological industry, as well as when observing the direction in which the world is heading with this regard, the importance of the mass of data cannot be denied. With trends such as Big Data [3] and the Internet of Things [4], the technological world has identified the significance and value that is possible with understanding and organizing the data that is constantly around us.

Newspaper articles have been a significant medium through which the common person has been able to receive information for hundreds of years. Through newspapers, people have received daily updates on the on goings of the world in all areas of interest. Any significant event that has happened during these years is likely to have been recorded in the format of newspaper articles. As the world has been shifting to a digital format, newspaper articles have also shifted to be distributed through the World Wide Web. The implications of this is that we are now able to process the data that newspapers hold in a much more efficient and accessible manner.

The research project that is outlined within this report was to implement a way in which newspaper articles could be grouped by their contents. The research involved that will be outlined within this report is primarily related to the areas of information retrieval and clustering. As the volume of text documents is constantly increasing in the world, the need to organize and group this information is only becoming more necessary [5, 6]. Additionally, when considered with the importance and significance that newspaper articles have, a plethora of reasons to pursue this research project can be identified.

This report is intended to outline and detail several agendas. Primarily it will showcase the web application developed through the course of this research project, clusterr. Various details and concepts that have been explored throughout the implementation of clusterr will be clarified, as well as details on the evaluation of clusterr. Finally, the various applications of clusterr will be explored as well as details on possible future work or refinements for the implementation of clusterr.

II. Implementation

A. Requirements

From the outset of the project, the initial specification was to provide a way through which newspapers could be grouped by the significant components of their contents. However, as this was an application that was to be used by people, there was heavy emphasis on the user experience from the start. Due to this, the way in which the user would be able to interact with the application and have an intuitive experience was held in high regard. As such, the requirements of the project could be summarized as such:

- To be able to group newspaper articles by their contents
- To allow a user to be able to access and read these newspaper articles
- Have the experience of the user be highly visual and interactive.

As this project was considered to be a proof of concept, the end user was envisioned to be that of just an ordinary person who desired to read newspaper articles and see what articles were related to those articles. Given that it was a proof of concept, this project was used to understand what kinds of ways newspaper articles could be grouped together, as well as how these groups or clusters could be formed. From this point on within this report the project will be referred to by its name, ‘clusterr’.

B. Preprocessing

Although as mentioned previously, a significant motivation for pursuing the accurate clustering of newspaper articles is within the fact that there are so many articles that are available to us, this also presents the problem of its own. Within all of this data that requires processing, there is also a significant amount that is not necessary when deciphering the contents of the documents. Due to this, a stage of preprocessing is required in which the data that we cluster is essentially filtered and optimized so that unnecessary processing is avoided, whilst also providing more accurate results [7]. The two main steps of preprocessing that is used within the implementation of clusterr are the filtering through the removal of stop words and optimization through the lemmatization of words.

1) Stop Words

Even when quickly scanning or reading through any document, it can easily be seen that there are some words that carry more significance to the meaning of the document than others, whilst some words carry no significance at all. A group of words called ‘Stop Words’ can be considered to be the latter [8], consisting of the most common words within the English language. In virtually all techniques that involve the processing of data concerning language, the text is filtered to an extent, in this case through stop words. Through this filtering, the presence of words that have no real significant value are removed. Some examples of stop words that have been filtered out are ‘the’, ‘it’, ‘which’, or ‘on’. As a result of the filtering of these words out of the documents to process, any unnecessary computation is avoided while giving a more accurate representation of the contents of the documents.

2) Lemmatization

Due to the way in which the English language is structured, it is easily seen that a direct analysis of the words within a document may give inaccurate results regarding frequencies of words. There are many different forms a single word can have, depending on the grammatical context that the word is placed in. For example, the connotation the words 'running, 'runner', and 'run' give are all the same and would signify the same meaning in a document but would be treated as different words. When considering this, it is highly likely that when processing the individual words within a document, that due to these different forms, a single word may appear to be less significant. Due to this problem, a process in which the base form of the words is found is needed.

There are two popular methods through which this is usually done, stemming or lemmatization [9]. Stemming is usually a very rudimentary process in which words that are placed through the algorithm are directly truncated. While this is a very simple approach, it does have many disadvantages, as it is a very crude process in which words may be truncated off into base forms that do not exist. For example, with the sample of using the words 'running', 'runner', and 'run', the word 'run' may be truncated down to just 'r', when placed through a stemming algorithm.

Lemmatization however is a little more complicated in that it involves the morphological analysis of words. By identifying and returning the base or dictionary form of a word, referred to as the 'lemma', this algorithm does a far more accurate job in identifying words with common meanings. For example, if the word 'run' were to be lemmatized, either 'ran' or 'run' would be returned depending on the context in which the word was being used.

Following extensive research into the benefits of these methods, it was decided that Lemmatization would be implemented within the preprocessing involved within clusterr.

C. Tf-Idf

As the contents of text documents are not necessarily quantifiable in its raw form, there is a need to convert it so that we are able to cluster and group the documents more easily. Essentially there is a need to convert the words within the documents into numbers. This is where the concept of Tf-Idf is used. Tf-Idf stands for Term Frequency - Inverse Document Frequency [12] and is essentially a value or a weighting that is used for each word in a document, giving an indication of how important that word is within a document, where the document is a part of a corpus of documents.

This weighting is governed by three factors. As the term occurs more in a small selection of documents the weighting rises as the term is determined to be rare. As such, the weighting then decreases as the number of occurrences of a term decreases or is seen to be present in many documents. Thirdly, the weighting is at its lowest when the term is common in virtually every document. By finding the terms of a document with the top Tf-Idf values, it is possible to understand the main topics that are covered within the document.

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t.$$

Equation 1 - Tf-Idf Formula

D. Clusters

In order to properly understand the various concepts regarding clustering algorithms that will be addressed, a simple overview of clusters is given. Essentially, within the types of clusters that are dealt with regarding articles in cluster, each article is to be plotted on a two-dimensional space. As is natural within a two-dimensional space, some articles are in a closer proximity to other articles, and due to this some articles can be seen to be grouped together. A group of articles is then considered to be a cluster, with the center of these clusters being regarded as the centroid. How these clusters are made are dependent on which of the various clustering algorithms are used.

1) Silhouette Value

When using clustering algorithms, it is important to be able to validate the clusters and centroids that have been found. One method of evaluation is to calculate the silhouette value of the nodes. The silhouette value is essentially a measure of how appropriately a node has been placed in a cluster [10]. The equation for this measure is as shown.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Equation 2 - Calculation of Silhouette Values

Here, *b(i)* can be understood as the distance between the relevant node *i* and the closest cluster that the node does not belong to, whilst *a(i)* can be understood as the distance between *i* and its own cluster's centroid. Through the calculation of the silhouette value through this formula, a value between -1 and 1 is returned, where a higher value indicates that the node is in the correct cluster. By performing this calculation on each of the nodes and finding the average silhouette value, it is possible to evaluate the accuracy or correctness of the overall clusters. While this is a very computationally intensive technique, it was determined during implementation that it the computational costs were outweighed by the benefits that it presented.

2) K-Means Clustering

The K-Means clustering algorithm is a common algorithm used by researchers regarding clustering [11, 12]. This is an iterative method that is often used due to its computationally fast nature as well as its simplicity. The way in which K-Means begins is by first specifying the number of clusters, *k* that the algorithm

will cluster the input data into. Once this has been done, it assigns *k* number of *means* onto the data domain. Once these means have been assigned, it repeats the following steps iteratively until changes can no longer be made:

1. Clusters are made by associating each of the nodes with its closest *mean*
2. The centroids of these clusters becomes the new *mean*

If the centroids of these clusters are already the new mean and no changes can be made, *convergence* has been reached and the final clusters have been found.

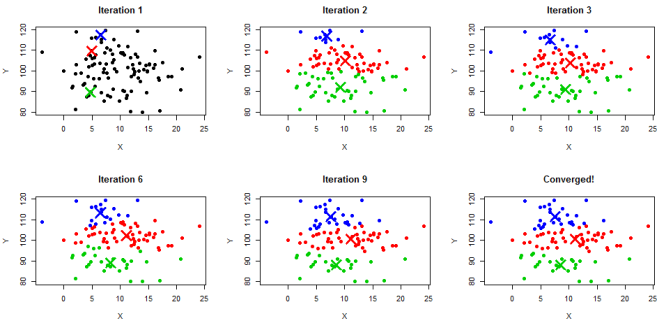

Figure 1 - Iteration steps of K-Means Clustering

3) Mini-Batch K-Means Clustering

The Mini-Batch K-Means clustering algorithm is an alternative to the ordinary K-Means algorithm that aims to reduce overall computational cost [14]. As mentioned previously, K-Means is an iterative process in which the entire dataset is used. Mini-Batch K-Means aims to reduce the computational cost of this iterative process by using fixed subsets of the data instead for large datasets. However, as this is an algorithm that is typically used for large datasets, it must be questioned whether the data that we are currently handling can be considered to be a large amount of data. This is further expanded upon in the evaluation section.

4) X-Means Clustering

As mentioned previously, one of the fallbacks of the K-Means clustering algorithm is within its requirement to specify how many clusters are needed. Due to this there have been several methods proposed in which the number k is automatically chosen. Methods in which this value is automatically chosen for k have been labelled as X-Means algorithms [13]. Some of the methods are:

1. Silhouette value evaluation
2. The Elbow Method
3. The Bayesian Information Criterion (BIC) Approach

The method chosen for the implementation of clusterr is through evaluating using the Silhouette value. Although the user still chooses a number of clusters when using clusterr, it is likely that this number that is specified is not necessarily the ideal number of clusters regarding the documents that are being clustered. Clusterr therefore iterates through each of the number of clusters up to the number specified and calculates the average silhouette value for each of these numbers and then chooses the number of clusters that returns the highest value.

5) Hierarchical Clustering – Agglomerative

Hierarchical clustering algorithms are methods that work in a way in which a hierarchical tree of clusters is created. This can be done in either a top-down or bottom-up fashion. Within the implementation of clusterr, only the bottom-up method, otherwise known as agglomerative clustering, is used. Within agglomerative clustering, it begins by assigning each document as its own cluster, and then successfully merging clusters to that which is closest to it [2].

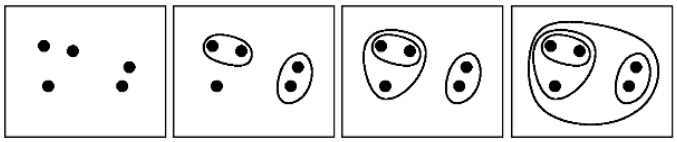

Figure 2 - Steps of Agglomerative Clustering

E. The Guardian API

In its current stage, clusterr retrieves its newspaper articles from the Guardian API [15]. This in itself brings about several implications. The first is that with the use of an API, there is now an inherent standardization of input that is being enforced on clusterr. This is due to the standard of the articles from the Guardian now being translated over to clusterr. However, this also means that the breadth of reach of cluster in its current stage is limited to that of articles available from the Guardian. This also presents opportunities of extensibility for clusterr however, as this API can easily be switched for another API, and with minimal adjustments, different articles are able to be retrieved.

F. Evaluation

1) K-Means Vs Mini-Batch K-Means

Whilst Mini-Batch K-Means is typically used as an alternative to K-Means as a way to decrease the computational time, this is typically used for large datasets. Additionally, it should also be understood that Mini-Batch K-Means typically tend to give lower quality clusters than a normal K-Means implementation.

Timed tests of clustering 100 articles and 200 articles show that when Mini-Batch K-Means is being used for datasets of this size, which is not typically considered large, Mini-Batch K-Means actually performs slower. For 100 articles it performed on average two times slower than an ordinary implementation of K-Means, and for 200 articles Mini-Batch K-Means performed on average approximately 1.6 times slower than K-Means.

Although this may indicate that in the current state of implementation, an ordinary K-Means may be more optimal, the kinds of data that clusterr is projected to handle are in numbers that are far greater than the limits that are placed by the API that it currently uses. Additionally, it should be noted that the typical lower quality clusters that result from Mini-Batch K-Means is usually not too disparate to that of K-Means.

Due to these reasons it was decided to continue to use Mini-Batch K-Means instead of K-Means within clusterr.

2) Agglomerative Vs X-Means

Throughout the implementation of clusterr it was clearly understood that both agglomerative and x-means clustering had its benefits as well as shortcomings. Depending on the dataset that was presented to these algorithms, they would perform differently. Due to the way in which these algorithms would act as well as the unpredictable nature, it was determined that including both of these algorithms within clusterr would be the idea decision. The duality of algorithms was included again through the use of silhouette values. Although this method adds to the computation within clusterr, it was regarded to be more beneficial to perform both clustering algorithms, calculate the overall silhouette value for each of them and then have clusterr evaluate and decide on the algorithm with the higher value, displaying the results of this clustering algorithm to the user.

III. Results

A. Interim Results

During the interim phase of the project, the system was able to produce a graph that consisted of just the plot values of each document. This was in many aspects very primitive and gave no real indication to the user regarding what the axis' represented or which plot was related to which document. Additionally, no clustering methods had yet been implemented and only gave the user a very minimal indication regarding the proximity of documents from each other regarding their contents. This current state satisfied none of the requirements that were stated previously and as such many changes were needed.

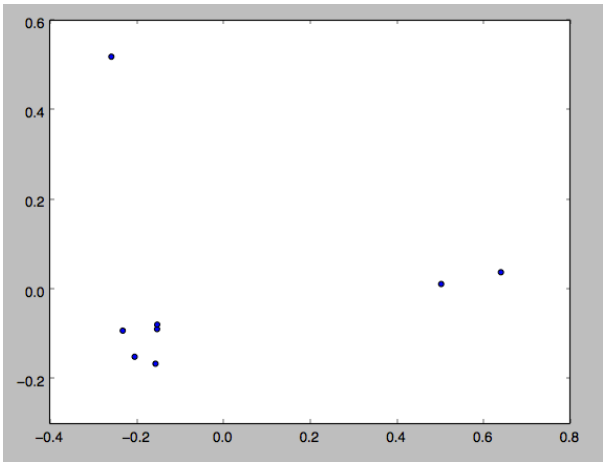

Figure 3 - Graph produced at Interim Stage

B. Final Results

In response to the lack of fulfilment of requirements at the interim stage, a web application named clusterr was built. This web application was designed with the user in mind, to be interactive, intuitive and aesthetic to use. It allows a user to be able to specify a number of articles, a date range and number of clusters, and once doing so it will retrieve the relevant articles through the Guardian API and begin clustering these articles using either the X-Means clustering algorithm laid over Mini-Batch K-Means or the Agglomerative clustering algorithm. Once this is finished it displays the results to the user in an interactive way in which the user is able to read the articles and visualize which articles are related to which.

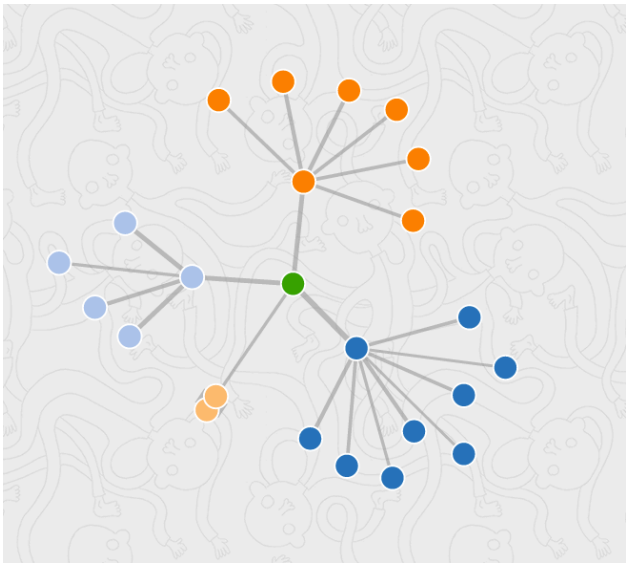

Figure 4 - Output of clusterr

C. Current Applications

Within the current stage of development that clusterr is in, it caters to the use case of an ordinary person who wishes to read newspaper articles. It is able to easily facilitate the viewing of articles for a user, whilst also showcasing visually the different articles that are available that share topics. However, even within this current stage of development, many more use cases can be identified, as virtually any form of significant text could potentially be used as input. For example, clusterr could potentially be used by students who are researching certain topics and wish to see what variety of resources are available that closely relates to the resources that they already have. Or extending upon this, where the current trend is for existing resources to move towards a digital format, a use case in which libraries employ an application such as clusterr to visualize their books could be seen as a possible application for clusterr.

D. Future Work

Within the current state of clusterr regarding the implementation, many areas of improvement and refinement have been identified. The first of these is within the method of choosing cluster numbers. To implement X-Means clustering over the Mini-Batch K-Means implementation, silhouette values are currently being used to evaluate the optimal number of clusters to choose. However, in the current implementation, the user still is required to specify a number of clusters, which acts as the maximum number of clusters that clusterr could choose to initialize. A suggestion for future work within clusterr is the implementation of the Bayesian Information Criterion,

which is a heuristic that allows the picking of an appropriate number of clusters.

Additionally, a further area of work that could be applied is within the lemmatization of words. To be able to use the semantic meanings of the words instead of their lemmatized forms is likely to be much more representative of the actual content of the documents, rather than a representative of the certain words the author chose to write.

IV. Conclusions

To conclude, through the progression of implementing a solution that groups together similar newspaper articles, many typical conventions and methods and information retrieval and clustering were researched. This involved a vigorous process of testing and validating each of these methods, weighing out the benefits of each method against the requirements set as well as the scope of the project. Concluding from these processes, we have presented clusterr, a web application that visualizes to the user a set of newspaper articles retrieved through the Guardian API, allowing them to see the contents of articles as well as the articles that are similar to it.

In its current stage of implementation, although clusterr was approached as a proof of concept, it is functional and can be seen as applicable in many areas with minor adjustments. Additionally, when considered with the current trends of the world regarding data, it can be seen that the ideas and concepts that clusterr is built upon are extremely valuable, supporting the various possibilities and futures of extensions of clusterr.

V. Acknowledgments

I would like to express my gratitude firstly for my partner, Ruoyi (Zoe) Cai, for her partnership and contribution to this work and direction. Furthermore, I would like to extend this gratitude to our supervisor, Gill Dobbie, for her constant guidance, direction and support throughout the course of this research project.

References

[1] Y. Zhong, D. Liu, *The Application of K-Means clustering algorithm based on Hadoop*. Cloud Computing and Big Data Analysis, July 2016.

[2] M. Al-Azawi, Y. Yang, H. Istance. *A New Gaze Points Agglomerative Clustering Algorithm and its Application in Regions of Interest Extraction.* Centre for Computational Intelligence, De Montfort University, 2014.

[3] B. M. Wilamowski, B. Wu, J. Korniak, *Big Data and Big Learning*. Department of Computer Engineering, University of Information Technology and Management, June 2016

[4] L. Wang, R. Ranjan, *Processing Distributed Internet of Things Data in Clouds*. IEEE Cloud Computing, 22 April 2015

[5] S. Padmaja, S. Bandu, S. Fatima, P. Kosala, M. C. Abhignya, *Comparing and Evaluating the Sentiment on Newspaper Articles: A Preliminary Experiment.* Science and Information Conference, August 2014

[6] A. Huang, *Similarity Measures for Document Clustering.* Department of Computer Science, The University of Waikato, April 2008

[7] A. I. Kadhim, Y. Cheah, N. H. Ahamed, *Text Document Preprocessing and Dimension Reduction Techniques for Text Document Clustering.* Department of Computer Science, College of Medicine. 2014

[8] List of Stop words in English Dictionary. May 2016. Available: http://www.ranks.nl/stopwords

[9] J. M. Torres-Moreno, *Beyond Stemming and Lemmatization: Ultra Stemming to Improve Automatic Text Summarization,* Laboratoire Informatique d'Avignon, September 2012

[10] S. Guido. (2014, Jun 13). *K Means Clustering with Scikit-learn* [Video File]. Retrieved from https://www.youtube.com/watch?v=-J9ZICyev5E

[11] J.Z.Huang, M.K.Ng, H.rong, Z.Li, Automated variable weighting in k-mean type clustering, IEEE Transactions on PAMI 27, 2005

[12] C. D. Manning, P. Raghaven, H. Schütze, *Introduction to Information Retrieval (Online Edition),* 2009

[13] D. Pelleg, A. Moore, *X-means: Extending K-means with Efficient Estimation of the Number of Clusters*, School of Computer Science, Carnegie Mellon University

[14] A. Feizollah, N. B. Anuar, R. Salleh, F. Amalina, *Comparative Study of K-Means and Mini Batch K-Means Clustering Algorithms in Android Malware Detection Using Network Traffic Analysis.* Computer System and Technology Department, University of Malaya, 2014.

[15] Guardian API Documentation. September 2016. Available: http://open-platform.theguardian.com/documentation/