Final report

Lee Chu Yong Mark, Sit Wing Yee

September 18, 2015

1 Introduction

In recent years, vast amounts of information have become available on the Internet. When an event occurs, information about this event is generated from numerous sources. However, events do not occur in isolation. More often than not, a single event can be connected to other related events in the past, present and future (unless of course, the said event is very recent). To a human analyst, it would be useful to be able to spot connections between events, which would allow them to gain additional insight to a single event or a greater understanding of a larger set of related issues.

In light of this, our objective is to develop a system that facilitates the discovery of relevant knowledge from a large number of news articles based on an initial topic of interest. This will be achieved through two goals:

1. Identify articles that report on the same topic
2. Chain connected topics that explain the development of a series of events

The aim of the first goal is to aid users in sifting through a large amount of information. This is accomplished through article clustering (section 3.3). The aim of the second goal is to facilitate the discovery of relevant knowledge by chaining the article clusters from the previous goal together using overlapping community detection (section 3.4).

The report is structured as follows:

- Section 2 - Summary of clustering methods and algorithms
- Section 3 - Implementation details
- Section 4 - Evaluation metrics and usage example
- Section 5 - Future work
- Section 6 - Related information

2 Background

Many authors have worked on the discovery of knowledge from news articles. We draw on work from two key areas, clustering for topic detection as well as overlapping community detection for creating chains of news articles.

2.1 Clustering

Clustering is the process of examining a collection of points and grouping these points into cluster according to a distance measure [1]. We elaborate on two clustering algorithms, hierarchical agglomerative clustering and KMeans. Both these algorithms assume an Euclidean space. Every cluster has a *centroid*, the average of all its points. The *radius* of a cluster is the maximum distance between any point and the centroid. The *diameter* of a cluster is the maximum distance between any two points in the cluster.

2.1.1 Agglomerative clustering

In agglomerative clustering, each point starts as its own cluster. Clusters that are close to each other, based on a distance measure, are combined to form a larger cluster. This process of combining clusters continues until certain conditions are met (e.g. k clusters are desired and have been found) or it becomes undesirable to combine two clusters together (e.g. the diameter of the best possible merge exceeds a threshold). Agglomerative clustering suffers from a running time of $O(n^2 logn)$, making it infeasible when n is large. This stems from having to compute the distance between each pair of clusters to find the best merge.

2.1.2 KMeans

Unlike agglomerative clustering, which merges clusters together, KMeans is a point clustering algorithm. It assumes that there are k clusters. To select these initial k clusters, k points are chosen from the data and set as the centroids of their clusters. These points are chosen in such a way that they are unlikely to be in the same cluster. For each of the remaining points, the centroid nearest to the point is found, the point is added to the cluster of that centroid and the centroid is adjusted to account for the addition of the point. KMeans runs in linear time with respect to the number of items.

2.1.3 Bisecting KMeans

Comparisons of clustering approaches have found that while agglomerative clustering tends to produce superior clusters, KMeans is far more efficient [2]. It is possible to combine the best of both these approaches and produce clusters which are as good or better than clusters produced by agglomerative clustering methods. The algorithm is discussed in detail in [2].

2.1.4 Clustering and topic detection

Clustering is used to detect topics. Columbia's Newsblaster [3], a news tracker and summariser, utilises agglomerative clustering with a group-wise average similarity function to group articles that belong to the same story together. In [4], an incremental clustering algorithm is proposed for topic detection.

2.2 Chaining algorithm

After generating clusters of articles, we need to link the clusters together to form a chain. We adopt the algorithm found in [5], which draws on the work in overlapping community detection from [6]. This algorithm captures the property of *coherence* effectively.

Coherence is the idea that every item in a chain should share some characteristics. When the items are article clusters, all the clusters should share a common set of words. While it is simpler to measure the similarity between consecutive clusters in a chain, it can give rise to chains that are incoherent. In an incoherent chain, all consecutive clusters are similar (e.g. clusters 1 and 2 are similar, clusters 2 and 3 are similar). However, clusters 1 and 3 may be very different from each other, to the point of not having any content in common. To apply the concept of coherence to the article clusters, we want to find groups of words that belong to the same clusters and clusters that use similar words [5].

A weighted bipartite graph is constructed, with the clusters as one set of nodes and individual words as the other. An edge exists between an article cluster and a word if and only if an article in the cluster contains the word. The weight of the edge is the number of times the word occurs in the cluster. By adding weights to the edges, it is possible to ensure strong co-occurrence between clusters and words by removing edges with weight that is less than 10% of the maximum edge weight.

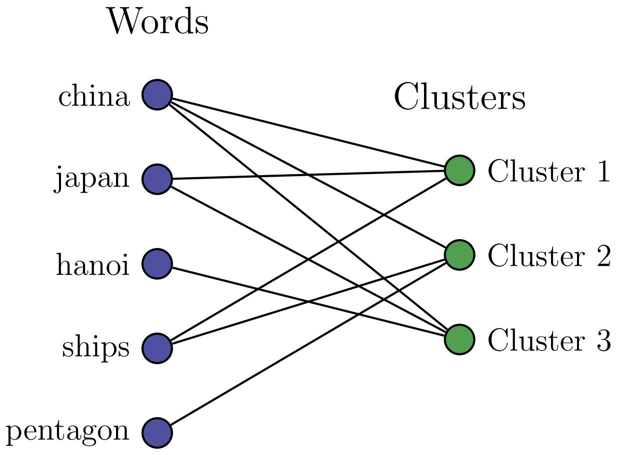

Figure 1: Sample bipartite graph after edge weights have been discarded

The weights are then discarded and overlapping communities detected using BigClam. BigClam uses a block coordinate gradient ascent approach and is very scalable, with each iteration taking near constant time. It is possible for a node to be in multiple overlapping communities. Further details on this algorithm can be found in [6].

3 Implementation

In this section, we provide a broad overview of our implementation, which has two main steps:

1. Group articles into article clusters - Bisecting KMeans
2. Chain article clusters together - Overlapping Community Detection

3.1 Data

We used the New York Times (NYT) API to load information on 208 259 articles from the U.S. and World sections from 2013 to June 2015, with 103 414 articles coming from the U.S. section and the remaining 104 845 from the World section. We captured the following information from each article: its title, the section which it came from (either U.S. or World), publication date, word count and a short summary. The data was stored in a SQLite database for further processing.

3.2 Tools

The entire project was written in Python 2.7.6 and numerous libraries provided additional functionality.

- **Gensim** - Dimensionality reduction, log-entropy model, general string processing, similarity interface
- **Scikit-learn stack** - KMeans clustering, mathematics
- **Snap-py** - Overlapping community detection
- **SQLite** - Database

3.3 Article clustering

3.3.1 Selection and pre-processing

The initial step involves splitting the articles into batches of 400 articles each. The title and summary of each article is concatenated into a single string. This string undergoes standard text pre-processing, which involves the removal of HTML tags, punctuation, multiple whitespaces, numbers and stopwords as well as the generation of bigrams and trigrams, before conversion into a bag-of-words (BOW) vector. The log-entropy model is applied to these BOW vectors to reduce the weight of words that are either rare or occur frequently. Log-entropy was chosen over other weighting models due to its excellent performance with latent semantic indexing over numerous data sets [7].

3.3.2 Dimensionality reduction

Dimensionality reduction is performed using latent semantic indexing (LSI). LSI is a method for automated indexing and retrieval that takes advantage of implicit higher order structure in the association of terms with documents. LSI was chosen over its successor, Latent Dirichlet Allocation (LDA) as it scales better and has greater noise tolerance. This reduces the number of dimensions from the number of unique words to the number of topics in the LSI space, which for this purpose is 50. We acknowledge that this is significantly lower than the norm for LSI (200 - 300). However, this is due to the nature of the data that we have gathered.

3.3.3 Clustering - Bisecting KMeans

In each batch of articles, it is common to find several articles covering the same news story. Articles which cover the same (or very similar) news story are grouped using bisecting KMeans. Bisecting KMeans is favoured over agglomerative clustering for reasons detailed in section 2.1.3. To reduce the size of the clusters, clusters containing more than 5 articles are split into smaller clusters. This size reduction is necessary to prevent the occurrence of large clusters, that are likely to contain articles which pertain to multiple stories or are extremely loosely related. Such clusters are not suitable for chaining.

After the clustering is complete, the contents of the articles (their titles and summaries) in each cluster are concatenated to form the content of the article cluster. Other relevant information (further details in the documentation) for each cluster is stored in a database for use in the next step.

3.4 Chaining article clusters

3.4.1 Similarity query

In section 3.3, clusters of articles were generated. As we have not developed a method to effectively discern between chains, we restrict the search space even further to facilitate subsequent evaluation. This restriction is necessary as it ensures that the clusters used in overlapping community detection are highly relevant to the query. If the clusters are irrelevant, the quality of the chains will suffer correspondingly. This restriction is performed using a similarity query.

The first step is to convert the contents of the clusters into a LSI space. The query is converted into the same LSI space as the contents. Dimensionality reduction is needed as at the end of section 3.3.3, the contents of the articles are concatenated to form a cluster and the content of a cluster is a long string. This reduces the number of dimensions for the clusters and significantly speeds up the search for clusters that are similar to the query, which is performed by cosine similarity.

3.4.2 Overlapping community detection

After finding the top N most similar clusters to the query, we then perform clustering using the algorithm described in 2.2. The value of N depends on how many communities are desired and the size of each community. Should i communities be desired, each containing an average of j clusters, then $N = i * j$. It is possible to adjust N, by restricting the number of input clusters as well as i, which can be specified to the community detection algorithm. Adjusting both values in tandem ensures that j is reasonable. For example, if j is too small, each chain would on average consist of one or two articles, which would not be useful. At the same time, if j is too large, the number of articles in the chain would be too large to be useful or the chain would contain a significant amount of irrelevant articles. The exact values of N and i, and by extension j, would be left to the user to decide depending on his or her usage case.

Every community that is detected contains one or more clusters, each of which contains one or more articles. All the articles in the community are then used to form a chain.

4 Evaluation

In this area of work, the principal evaluation technique is user testing [3, 5, 8, 9, 10]. Due to time constraints, we were not able to conduct user testing and instead we decided to use other metrics to evaluate the quality of our clusters as well as the quality of the chains.

4.1 Cluster quality

Radius and diameter are used to evaluate the quality of the article clusters. It is then necessary to adjust the earlier definitions of centroid, radius and diameter to fit the context of article clustering. The centroid is the article that best represents the cluster. The radius of the cluster is the maximum Euclidean distance between any article and the centroid. The diameter of the cluster is the maximum Euclidean distance between any two articles in the cluster.

Given these definitions, the ideal cluster would have a low radius and diameter, which would indicate that all the articles in the cluster are largely similar and should pertain to a single issue. It is possible to get a cluster of reasonable quality with low radius and high diameter, which would point to the cluster having one or two articles that are irrelevant.

4.2 Chain quality

To assess the quality of chains, a definition of coherence from [8] was used, allowing rapid computation of the quality of each chain. The concept of coherence was discussed in section 2.2 and for evaluation purposes, it is defined as follows:

$$Coherence(c_1...c_n) = \underset{i=1...n-1}{min} \sum_{w} 1(w \in c_i \cap c_{i+1}) \tag{1}$$

For every pair of adjacent clusters in the chain, the number of words that they share from the set of words in the entire corpus, w, is computed. Each word they share adds one to the similarity score between them. The coherence of the chain with clusters c_1 to c_n is the minimum of the similarity score between any pair of adjacent clusters in the chain.

An ideal chain would have high coherence, indicating a high degree of similarity between even the pair of adjacent clusters that are least similar to each other.

4.3 Usage case: South China Sea

We ran our evaluation metrics on a subset of the data gathered, attempting to find chains of events pertaining to issues occurring in the South China Sea. By initially filtering the raw articles to only include those with the keyword ”china”, we were left with a subset of ∼6000 articles (section 3.3.1). After performing dimensionality reduction (section 3.3.2), clustering using bisecting KMeans resulted in ∼1700 clusters (section 3.3.3). Clustering was carried out using overlapping community detection as well, which produced ∼2000 clusters (section 3.4.2, replace clusters with articles).

Table 1: Average diameter and radius for different cluster sizes

Cluster size		**2**	**3**	**4**	**5**
Average diameter	KMeans	0.84	1.09	1.14	1.16
	Community detection	1.04	1.12	1.14	1.16
Average radius	KMeans	0.42	0.64	0.70	0.73
	Community detection	0.52	0.65	0.70	0.74

In the table 1, we detail the average radius and diameter for the clusters of size 2 to 5 produced by bisecting KMeans and overlapping community detection. Although the results are similar, it is important to note that while 99% of clusters from bisecting KMeans have size of 2 to 5, only around 50% of clusters from overlapping community detection fall within the same range. As the sizes of the clusters have been forced downwards when using bisecting KMeans, there would be instances where splitting the cluster was not favourable. The actual scores for overlapping community detection would be higher if the majority of clusters were within the size range.

Both sets of clusters were then used to form chains using the query ”south china sea” (section 3.4). The similarity query returned 50 clusters and we picked what we felt were the best chains from both sets of clusters and evaluated their coherence (section 4.2). We list the chains in the tables below, displaying the articles closest to the centroid of each cluster:

Table 2: Chain produced by clusters created using bisecting KMeans, **Coherence: 7**

Headline	**Date**
Philippines protests Chinese use of water cannon	25 Feb 2014
China rebuffs US efforts on South China Sea tensions	10 Aug 2014
Vietnam calls for self-restraint in disputed South China Sea	17 Mar 2015
Chinese president promotes regional vision at Boao forum	27 Mar 2015
US hopes Chinese island-building will spur Asian response	28 May 2015
EU, Japan wary of unilateral actions in South China Sea	29 May 2015
China, US tone down rhetoric but far from South China Sea solution	31 May 2015

Table 3: Chain produced by clusters created using overlapping community detection, **Coherence: 9**

Headline	**Date**
Vietnam: warning issues to China over oil rig in disputed waters	7 May 2014
Vietnam: Chinese ships ram vessels near oil rig	7 May 2014
Philippines Aquino says China violates informal code at sea	19 May 2014
Hanoi says Chinese boat sinks Vietnamese fishing vessel in disputed waters	26 May 2014
China takes dispute with Vietnam to UN	9 Jun 2014
China says told Vietnam to stop hyping up South China Sea oil rig row	18 Jun 2014
China tells Japan to set down historical baggage	8 Mar 2015
China says progress being made on India border talks	8 Mar 2015
Vietnam calls for self-restraint in disputed South China Sea	17 Mar 2015
China peeved as Hillary Clinton denounces womens detention	7 Apr 2015

From tables 2 and 3, it is evident to us that the chain produced when article clustering was done by bisecting KMeans is more coherent than the chain produced when article clustering was done by overlapping community detection, even though the former had a lower coherence score. This further underscores the importance of user testing in this area of work.

5 Future work

This section contains information that we feel would definitely be useful for further work on this topic. However, there was insufficient time and / or domain knowledge to consider the approaches mentioned in this section in detail.

5.1 Automatic chain evaluation

Shahaf proposes a method for automatic chain evaluation in [5]. This method involves optimising a submodular function and was not investigated in detail due to a lack of domain knowledge.

5.2 Improving cluster quality

An easy way to improve the quality of the chains would be to improve cluster quality. We suggest several possible ways to improve cluster quality.

doc2vec: This is an enhancement of the work in [11]. It is a new method of dimensionality reduction that utilises deep learning to capture the semantic meaning between words. An efficient implementation is provided in Gensim and further testing should be done to determine if its performance is superior to a combination of log-entropy weighting and latent semantic indexing. If doc2vec is able to provide better differentiation between articles after dimensionality reduction, it should result in higher quality clusters being produced.

Custom weighting: The current dimensionality reduction process does not fully take into account the importance of certain words in news articles. For example, words associated with names and places should be viewed with greater importance. By utilising a natural language parser (Stanford parser), these words can be identified and their weight raised, leading to better differentiation between articles and in turn higher quality clusters.

Supervised learning for clusters: An observation from our testing was that it was easy for a human observer to differentiate between clusters that were relevant and those that were not. Should a human observer tag a sufficient amount of relevant and irrelevant clusters, it is possible to perform supervised learning on the clusters and include other features such as the number of articles in the cluster, the length of the articles in the cluster, the number of comments each article in the cluster received and so on. These additional features may be useful in differentiating between relevant and irrelevant clusters. This approach would improve cluster quality by eliminating low quality clusters.

5.3 Multiple data sources

Articles were only drawn from a single data source, the New York Times. Drawing articles from multiple sources will allow for additional insights as news sources often differ in their coverage of the same event. For example, it is unlikely that the New York Times, being based in the United States of America (USA), would portray the USA in an extremely poor light. However, a news source from China may disagree and describe the USA negatively. It would then be possible to create chains on the same issue that differ in tone and / or opinion, a useful feature for analysts.

6 Related information

This section contains other information which may be useful for further work on this topic but was not directly used in the project.

6.1 Alternative article clustering method

The algorithm in section 2.2 is an extension of an algorithm used to cluster articles together. It entails creating a word co-occurence graph for a set of articles. In such a graph, words are nodes and edges are created between the nodes if they occur in the same document. The top 50 tf-idf words were used for each article [5]. Bigclam is used to find overlapping communities in the graph [6]. This results in clusters of words and it is unclear to us exactly how these word clusters are linked to the original articles, making it impossible to create article clusters for subsequent analysis. A possible method of doing so would be to calculate the similarity between the word clusters and the top 50 tf-idf words from each article. If the similarity exceeds some threshold, the article is linked to that word cluster.

6.2 Temporal summarisation

Temporal summarisation systems aim to monitor information associated with an event over time [12]. Work in this area aims to create systems that can provide relevant updates as a situation progresses and tracks important event-related attributes. When used to examine a single topic, it can yield greater insight into how events unfolded. Unfortunately, it requires that the topic be identified before it can be utilised [13].

References

[1] A. Rajaramna and J. D. Ullman, *Mining of massive datasets*, vol. 77. Cambridge University Press, Cambridge, 2012.

[2] M. Steinbach, G. Karypis, V. Kumar, *et al.*, "A comparison of document clustering techniques," in *KDD workshop on text mining*, vol. 400, pp. 525–526, Boston, 2000.

[3] K. R. McKeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, J. L. Klavans, A. Nenkova, C. Sable, B. Schiffman, and S. Sigelman, "Tracking and summarizing news on a daily basis with columbia's newsblaster," in *Proceedings of the second international conference on Human Language Technology Research*, pp. 280–285, Morgan Kaufmann Publishers Inc., 2002.

[4] K. Rajaraman and A.-H. Tan, "Topic detection, tracking and trend analysis using self-organizing neural networks," in *Advances in Knowledge Discovery and Data Mining*, pp. 102–107, Springer, 2001.

[5] D. Shahaf, J. Yang, C. Suen, J. Jacobs, H. Wang, and J. Leskovec, "Information cartography: creating zoomable, large-scale maps of information," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1097–1105, ACM, 2013.

[6] Y. Jaewon and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach," 2013.

[7] T. K. Landauer, D. S. McNamara, S. Dennis, and W. Kintsch, *Handbook of latent semantic analysis*. Psychology Press, 2013.

[8] D. Shahaf and C. Guestrin, "Connecting the dots between news articles," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 623–632, ACM, 2010.

[9] R. Nallapati, A. Feng, F. Peng, and J. Allan, "Event threading within news topics," in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pp. 446–453, ACM, 2004.

[10] D. Shahaf, C. Guestrin, and E. Horvitz, "Trains of thought: Generating information maps," in *Proceedings of the 21st international conference on World Wide Web*, pp. 899–908, ACM, 2012.

[11] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *arXiv preprint arXiv:1405.4053*, 2014.

[12] J. A. Aslam, M. Ekstrand-Abueg, V. Pavlu, F. Diaz, R. McCreadie, and T. Sakai, "Trec 2014 temporal summarization track overview.," in *TREC*, 2014.

[13] J. Allan, R. Gupta, and V. Khandelwal, “Temporal summaries of new topics,” in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 10–18, ACM, 2001.