Progress report

Lee Chu Yong Mark

August 18, 2015

Abstract

Placeholder

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). It may be difficult for human analysts to spot these connections between events.

TBC - placehold

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 all points 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 unfeasible 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

Comparions 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 [2] but not utilised due to limited time.

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 groupwise 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 Article chaining

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-occurence between clusters and words by removing edges with weight that is less than 10% of the maximum edge weight.

The weights are then discarded and overlapping commmunites detected using BigClam. BigClam uses a block coordinate gradient ascent approach and is very scalable, with each iteration taking near constant time [6].

3 Algorithm

In this section, we provide a broad overview of our algorithm. Our algorithm has two main steps.

1. Group articles into article clusters - Agglomerative Clustering
2. Chain article clusters together - Overlapping Community Detection

3.1 Article clustering

The initial step involves splitting the articles into time steps of seven days. Articles are vectorised using TF-IDF. TF-IDF aids in the removal of uninformative words by reducing the weight of words that are either rare or occur frequently.

Latent Semantic Indexing (LSI) is performed. LSI is a method for automated indexing and retrival 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.

As we examine many articles at each time step, 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 agglomerative clustering. Agglomerative clustering is favoured over KMeans as it results in smaller clusters that more accurately capture a single story. To further reduce the size of the clusters, clusters with more than 10 articles are split into smaller clusters. This has the added benefit of ensuring consistent cluster size, as certain time steps contain significantly more news articles than others.

3.2 Chaining articles

In section 3.1, we have generated clusters of articles. To form chains of articles, the algorithm described in section 2.2 was applied on the clusters. Every overlapping community that is detected may contain one of more clusters, each of which contains one or more articles. All the articles in each overlapping community are used to form a chain.

4 Evaluation

4.1 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.

4.2 Metrics?

This section is still very open for discussion. The most common metric seems to be user testing on a specific topic.

4.3 SQL DB

- NewYT all (article) - name text, section text, date datetime, wordcnt int, summary text, id int
- NewYT clustered (cluster) - content text, n articles int, first article date datetime, last article date datetime, articles id text (json), articles data text (json), id int

5 Others

Have not decided where these might go, but the information here could be useful if anyone else decides to examine this area in the future.

5.1 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. 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.

5.2 Temporal summarisation

Temporal summarisation systems aim to monitor information associated with an event over time [7]. 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 [8].

6 Progress

6.1 Complete

- Clean raw articles and put into sqlite DB
- Convert cleaned articles to vectors using LSI topic model, articles are processed in batches of 1 week each
- Agglomerative clustering on these vectors
- Expand largest clusters as needed
- Some error occur while performing clustering. Remove articles that cause errors and try again / save the cluster
- Save these clusters as the top articles
- Store the top articles into some DB as well
- Convert top articles into vectors (LSI)
- Process these using some algorithm to make the chains. Something simple will suffice for now, just to get something out to examine. Toposort?

6.2 To do - short term

- Writeup on the background

6.3 To do - long term

- Come up with a better way to determine whether a saved cluster (top article) is accurate - a possible solution might be supervised learning, come up with a tool to aid in the annotation
- Switch tfidf to use logent weights
- Improvements on the chains of articles?

7 Reference list

- Mining massive dataset [1].
- Newsblaster [3].
- Contextual bandit [9].
- Clustering survey [10].
- Clustering technique comparison [2].
- Online LDA [11].
- TDT, trend analysis with neural network [4].
- TREC temporal summarisation track overview. [7]
- Topic detection and tracking evaluation overview [12]
- Connecting dots between news articles [13]
- Trains of thought: generating information maps [14]
- Event threading within news topic [15]
- Information cartography [5]

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 non-negative matrix factorization approach," 2013.

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

[8] 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.

[9] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*, pp. 661–670, ACM, 2010.

[10] P. Rai and S. Singh, "A survey of clustering techniques," *International Journal of Computer Applications*, vol. 7, no. 12, pp. 1–5, 2010.

[11] L. AlSumait, D. Barbará, and C. Domeniconi, "On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking," in *Data Mining, 2008. ICDM' 08. Eighth IEEE International Conference on*, pp. 3–12, IEEE, 2008.

[12] J. G. Fiscus and G. R. Doddington, "Topic detection and tracking evaluation overview," *Topic detection and tracking*, pp. 17–31, 2002.

[13] 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.

[14] 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.

[15] 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.