Detection of Topic Structure of the Guardian's News Webpages Based on Network Analysis

By Patorn Utenpattanun

Supervisor: Dr. Shi Zhou

This report is submitted as part requirement for the MSc in Web Science & Big Data Analytics at University College London. It is substantially the result of my own work except where explicitly indicated in the text.

September 2016

Abstract

The detection of topics in a document collection is an area of active research in the field of natural language processing. While most of the studies have been based on the analysis of text content, only a few studies undertake the problem by the analysis of a network. From the perspective of network analysis, these documents carry meaningful information in the connectivity between them which are valuable in a wide range of applications. This report explores a new approach to the problem of identifying the topic structure in a collection of the Guardian's news articles collected over one month period. Our approach is not based on text analysis but mainly grounded on the analysis of the graphical structure of a network. Specifically, we try to detect the community structure in a document graph. The result shows that the quality of topics and their associated topic words detected from our approach is as good as a standard textual method such as Latent Dirichlet Allocation (LDA). This result is encouraging since it confirms the value of the connectivity between documents that should not be ignored. We are not trying to replace textual approaches but we would like to point out that network analysis could offer an improvement to the textual approaches for the detection of topic structure problem.

Acknowledgements

I would like to thank my supervisor, Dr. Shi Zhou for his dedicated support, encouragement, and useful suggestions throughout this MSc project.

Finally I would like to thank my parents for their support throughout my study.

Contents

1 Introduction **1**

1.1 Report Structure 2

2 Background **3**

2.1 Detection of Topic Structure 3

2.2 Clustering and Cluster Labeling 4

2.2.1 Clustering 4

2.2.2 Cluster Labeling 4

2.3 Topic Modeling 5

2.4 Network Analysis 6

2.5 Evaluation Metrics 7

2.5.1 Cluster Quality 7

2.5.2 Topic Word Quality 7

3 Related Works **8**

3.1 Detection of Topic Structure based on Textual Approaches . . 8

3.2 Detection of Topic Structure based on Network Analysis . . . 9

4 Project Plan 11

4.1 Problem Statement 11

4.2 Objective 12

4.3 Dataset 12

4.4 Methodology 13

4.4.1 Clustering Methods 13

4.4.2 Topic Modeling 15

4.4.3 Network Analysis 17

4.5 Tasks 20

5 Task 1: Cluster Quality Comparison 21

5.1 Experiment Setup 21

5.1.1 Dataset 21

5.1.2 Tools and Settings 22

5.2 Results 24

5.2.1 The Optimal Threshold of Weighted Edges 24

5.3 Evaluation for Cluster Quality 24

5.3.1 Qualitative Evaluation 25

5.3.2 Quantitative Evaluation 27

6 Task 2: Topic Word Quality Comparison 30

6.1 Experiment Setup 30

6.1.1 Dataset 30

6.1.2 Tools and Settings 31

6.2 Results 31

6.2.1 Louvain Modularity Results 31

6.2.2 LDA Results 35

6.3 Evaluation of Topic Words 36

6.3.1 Qualitative Evaluation 36

6.3.2 Quantitative Evaluation 37

7 Discussions **40**

7.1 Summary of Experiments 40

7.2 Intuitive Explanations of Our Performance 40

7.3 Limitations and Improvements 41

7.4 Implication of Our Work for Topic Modeling 42

7.5 Future Works 43

8 Conclusion **45**

8.1 Summary of Works and Achievements 45

8.2 Self-evaluation 46

8.3 Final Notes 46

A Source code 47

A.1 Initial setup 47

A.1.1 Github 47

A.1.2 Dataset 47

A.1.3 Requirements 48

A.2 Manual 48

Chapter 1

Introduction

While more and more documents are available online in the present time, it becomes difficult to find the relevant documents; therefore, it would be an interesting idea to figure out a way to reorganize these document collections. One of the solutions is to group similar documents into topics and label the topics with relevant words.

Although this problem of a given text collection is well studied in the field of natural language processing, only a few studies undertake the problem from a network perspective. We propose a novel approach based on network analysis to solve the problem of finding topic structure in a document collection.

The motivation behind our approach is the profound nature of documents as a network. The documents in the network carry meaningful content in their connectivity which can be used for many applications. In real life, a variety of semantics can be used to link two documents together. Academic papers can be linked together via citations. Webpages can be linked together via hyperlinks. Intuitively, documents with similar contents can be linked together via content similarity.

In this work, we conducted experiments to prove our points by evaluating the performance of our approach on two tasks. The first task is designed to evaluate the clustering performance. We pick the standard clustering methods, which are K-means and hierarchical clustering, to compare with our network analysis approach. The second task is the topic coherence comparison between our network analysis approach and the textual approach which is latent dirichlet allocation (LDA).

The results show that our approach can detect better clusters than the other methods. In addition, the topic words extracted from the clusters of our approach are as coherent as the result from LDA. These results are motivating and they confirm the importance of the connectivity between documents. It would be interesting for further works to utilize network analysis as an enhancement to textual approaches in the problem of topic modeling.

The project makes the following contributions:

- We demonstrate that our network analysis approach can achieve a competitive performance on the detection of topic structure problem compared to the gold standard method namely latent dirichlet allocation (LDA).
- Our experiments are performed on the newly created document collection, the Guardian's news dataset, collected over one month period.

1.1 Report Structure

The remaining chapters of this report are organized as follows. Chapter 2 provides background materials on the detection of topic structure. The related methods and techniques will be briefly discussed. Chapter 3 reviews the related works to give readers the better understanding on the subject. The project plan is explained in Chapter 4. It outlines the project objective and the design of the experiments. Chapter 5 and Chapter 6 provide the detailed report on the clustering performance and the quality of topic words of our approach. Chapter 7 contains the summary of key findings and discusses the future works. Chapter 8 concludes this report.

Chapter 2

Background

The primary focus of this chapter is to give the reader background materials on the detection of topic structure.

2.1 Detection of Topic Structure

When there is a large collection of unstructured texts, an automatic tool is needed to organize the collection. The detection of topic structure problem can be described as identifying the hidden thematic structure in a collection of documents. In other words, the problem can be solved by finding document clusters and identifying words that thematically represent the clusters.

The detection of topic structure problem is formally formulated as follows. There is a collection of documents, $D = d_1, ..., d_n$, called a corpus, and the set of distinct words or terms $V = t_1, ..., t_n$ represents the vocabulary of D. A document is then represented as a n-dimensional vector $\boldsymbol{d}$ for n number of terms. With a document clustering method, we assign a document d into the most likely cluster C. The result of this task is a set $C = C_1, ..., C_n$ of clusters. Then, we find the semantic topic words for each cluster. The term topic words refers to a set of words within a cluster $W = t_1, .., t_n$. With a topic word extraction method, we extract a set $T = t_1, ..., t_n$ of topic words that most likely represent a given cluster.

2.2 Clustering and Cluster Labeling

The detection of topic structure problem can be undertaken by two processes, which are clustering and cluster labeling.

2.2.1 Clustering

Clustering is a data exploratory analysis task that aims to find the structure from a set of data points [**?**]. Its operation is to partition the data points into homogeneous clusters based on the values of the data. In most cases, a similarity metric is employed to measure similarities between different data objects. When they are deemed as similar objects, they are grouped together to form clusters.

Clustering is not a classification task because clustering is unsupervised learning of which category labels are not needed to determine clusters. Some of the popular clustering methods are as follows: K-means, hierarchical clustering, and spectral clustering. Clustering methods have been applied in a variety of disciplines including medicine, biology, social science, machine learning, information retrieval, and data mining.

2.2.2 Cluster Labeling

Cluster labeling is the process of giving labels to clusters [**?**]. Cluster labeling is important because people can see what a cluster is all about. The intuition behind this technique is that the terms having very low frequency cannot represent the whole cluster and cannot be used in labeling a cluster.

The commonly used approach is differential cluster labeling which selects cluster labels by comparing the distribution of terms in one cluster with the terms from other clusters. This approach uses feature selection techniques such as point-wise mutual information to select terms for cluster labels.

The other approach is cluster-internal labeling. The cluster-internal labeling identifies the labels based on the cluster itself. There are several ways to determine the cluster labels. For instance, the document title that is closest to the centroid of a given cluster can be used as a label.

2.3 Topic Modeling

Topic modeling is a suite of algorithms that aim to discover thematic topics from a document collection [**?**]. The algorithms analyze words in a document collection to identify the topics without labels for documents. The assumptions of topic models are that a document is consisted of multiple topics and a topic is represented by a probabilistic distribution over words.

Technically, for the input, a set of documents is transformed into a bag-of-words which is a sparse vector of word occurrence counts. Then the topic model processes this bag-of-words to generate a set of topics and a set of probability distributions over topics for each document as the output. Specifically, each topic is represented by a probability distribution over the vocabulary in the collection. The most representative words of each topic are the ones with the highest probability in the topic.

The definition of detection of topic structure may look similar to topic modeling. However, as described above, topic modeling goes beyond the scope of the detection of topic structure problem by modeling topic distributions over each word and topic distributions over each document out of a corpus. Instead of using topic modeling for data exploratory, the results of topic modeling are also used for dimensionality reduction which serves as the input for further data mining tasks such as information retrieval and classification.

2.4 Network Analysis

Network analysis applies graph theory to the analysis of complex networks. Graph theory is the study of graphs that represent the relationships among discrete objects [?]. A graph is defined as a collection of objects called nodes or vertices (V) where some pairs are connected together by links or edges (E). Mathematically, A graph can be represented by G(V,E) where E corresponds to the non-zero elements in the V x V adjacency matrix.

One of the active research areas of network analysis is the detection of community structure. The community detection algorithm invented by Girvan and Newman is the foundation of its variants in this category and it is proved to be highly effective at discovering community structure [?]. The problem of detecting communities can be referred to the clustering problem in data mining.

The Girvan and Newman algorithm revolves around the notion of edge betweenness. The edge betweenness of an edge i is the number of shortest paths between pairs of nodes that run along them. The algorithm involves calculating the edge betweenness of all edges in the network, removing the edge with the highest betweenness, and repeating this process until no edges remain.

Later on, Girvan and Newman propose the notion of modularity as a property of a network to determine how many communities a network should be split into [?]. The intuition behind modularity (Q) is that there should be many edges within communities and only a few between them. In practice, the modularity value is typically in the range between 0.3 and 0.7.

Since then, there is a demand to process very large networks in which we need algorithms with less computational complexity. In recent years, Louvain modularity [?] and Infomap [?] are the two recognized algorithms that satisfy the demand and they also give highly accurate results [?].

2.5 Evaluation Metrics

2.5.1 Cluster Quality

The objective of topic structure detection is to retrieve the groups of semantically coherent documents. The contents of documents in the same group should be similar and relatable. Internal and external metrics are the two types of evaluation metrics. The internal metrics are used when there is no ground-truth [?]. If we do not have the predefined number of topics, we use the internal metrics such as intra-cluster similarity to measure the similarity of all data points within a cluster. In contrast, external metrics are used when data are labeled [?]. These metrics compare the results with the true labels. Generally, the external metrics include F-measure, normalized mutual information (NMI), purity, and entropy [?].

2.5.2 Topic Word Quality

A set of topic words is the one that represents a particular topic. The good topic words should be understandable and interpretable by humans. In recent years, the recognized category of the metrics that are used to measure the topic quality is automatic topic coherence evaluation. The intuition behind this evaluation is that coherent topics will have topic words that frequently co-occur across the corpus. Newman et al. [?] created UCI coherence and conducted a comparison between their model and human evaluation and found that the automatic evaluation using a Wikipedia corpus can achieve nearly perfect agreement with human evaluation. Röder et al. [?] benchmarked the commonly used evaluation metrics including UCI [?] and UMass [?] and also proposed the new model, C_V as the refined metric that highly resemble human judgment.

Chapter 3

Related Works

3.1 Detection of Topic Structure based on Textual Approaches

Document clustering and topic modeling are two intensively studied areas in the text domain [? ?]. The great success in these areas has led to a wide range of applications. Document clustering helps organizing similar documents into groups, which enhances information-related tasks such as document organization, browsing and retrieval. Topic modeling helps to discover topics that best describe the given document collection and has shown its effectiveness in analyzing texts for data mining tasks.

Popular clustering methods such as K-means, hierarchical clustering, and spectral clustering in general clustering literature have shown good performance in document grouping [? ?]. Specifically to text clustering, the frequently used techniques that have applications in clustering tasks are latent semantic indexing (LSI) [?] and non-negative matrix factorization (NMF) [?]. These factorization-based techniques provide dimension reduction which reduces the noise of similarity measure and enhances the important semantic features in the underlying data [?]. This technique can also be used with the clustering methods. For example, after applying NMF to documents, K-means is used to find the final set of clusters from the transformed data.

Although the clustering methods can provide clusters from a document collection, they rely on the additional cluster labeling techniques to determine the sets of topic words that represent each cluster. In contrast, topic models are developed to provide a probabilistic approach to directly reveal the hidden topic structure. The first attempt to develop a topic model is probabilistic latent semantic analysis (PLSA) [**?**]. However, the model has a serious overfitting issue with a large text corpus and it does not give a probabilistic structure at the document level.

As a result of advancement in the topic modeling research, the current gold standard is latent dirichlet allocation (LDA) [**?**]. LDA captures exchangeability of words and documents that the previous techniques such as PLSA could not handle. In addition, the development of the efficient inference procedure of LDA has also been the main driver behind the popularity of LDA. For instance, the online variational inference of LDA has demonstrated the massive reduction in estimation time which allows the topic model to process a larger text corpus [**?**].

3.2 Detection of Topic Structure based on Network Analysis

Despite the popularity of topic models in the field of natural language processing, there are only a few research studies that apply a network analysis approach to the detection of topic structure problem. Most of the network analysis studies usually focus on the detection of important words from a word co-occurrence network. Although there have been reported successes with this approach, none has achieved the level of success as LDA in recent years.

The work of Matsuo et al. [**?**] shows that a graph of words and Newman's community detection method can be used to find the word groups in a document collection. The problem of this model lies in the process of building a graph where thousands of search queries are performed to obtain a co-occurrence matrix for each pair of words. Although the result is insightful, this method is impractical for a large text collection.

Grineva et al. [?] performed the keyword extraction by using the Girvan-Newman algorithm on a semantic word graph where each node is a word and each edge is the semantic similarity between two terms. Their results show that their approach can find high-quality groups of keywords that are comparable to the ones produced by standard algorithms such as TextRank [?] and Wikify [?].

Pivovarov and Trunov [?] applied the modified version of the Girvan-Newman community detection algorithm on a word-document bipartite graph to perform document clustering. They compared the method with other algorithms such as Spectral clustering [?] and Non-negative matrix factorization [?] and found that their method can achieve a competitive quality of the clustering.

The noteworthy model that combines network analysis with topic modeling is relational topic model (RTM). The creators of RTM were David Blei, the co-inventor of LDA, and Jonathan Chang [?]. This model incorporates the topic probabilistic distributions and the links between documents to estimate topics. The improvement in the performance compared to the original LDA indicates that the integration of network data is beneficial for topic modeling.

Chapter 4

Project Plan

4.1 Problem Statement

We are facing an age of unprecedented amount of digitized documents that has been collected and stored every single day. An emerging problem is that it becomes more difficult to retrieve the relevant documents.

Currently, the two main tools that we use to explore online information are search and links. We type the desired keywords into the search engine and obtain the keyword-related documents. We navigate from one webpage to another one by clicking on a link. However, the obstacle with these mentioned tools is that most of the users rarely know what they are looking for because they are unaware of the availability of a document.

Therefore, it would be more efficient if we can examine the documents based on themes or topics. Instead of solely depending on finding documents from keywords, we scan through the document collection by indicating the interested theme and explore the documents that are related to that particular theme.

4.2 Objective

As more and more documents are available online, it is impossible for human power to examine all the information. The objective of this project is to provide a novel approach to the problem of finding common and relevant topics in a collection of varied documents.

The detection of topic structure problem can be simply considered as identifying topics in the corpus and labeling topics with the words semantically related to the topics. For each document collection, the main task is to reveal topics that governing the collection. For each topic, topics can be labeled by the most representative words.

In this project, a network analysis approach will be used to build a network-based model for the detection of topic structure. The models will be applied to the Guardian's news corpus and evaluate the performances with the popular textual analysis such as latent dirichlet allocation (LDA).

While a network analysis approach has been used in a variety of issues such as social network analysis in recent years, a few research has attempted at discovering the topic structure. In order to showcase the importance of network connectivity, this project will employ the common and simple methods from network analysis instead of applying sophisticated algorithms.

4.3 Dataset

The corpus of news articles from the Guardian's open platform will be used in this study [**?**]. The open platform is a public service for accessing all the content of the Guardian publisher. One month period of data in May 2016 are retrieved from the platform. Since there is a limitation placed on the number of documents to be retrieved per request, we create a script to pull all articles. The total number of documents in the original dataset is 9,095.

4.4 Methodology

4.4.1 Clustering Methods

There are two standard clustering algorithms used here to find clusters from the document collection, which are K-means and hierarchical clustering.

4.4.1.1 K-means

Partitional clustering is a popular technique in document clustering. The partitional clustering creates a one-level or flat partition of the data points. The commonly used method for this technique is K-means [**?**]. This method is often brought as a baseline in document clustering literature.

K-means for document clustering can be formally described as follows. There is a set of n-dimensional document vectors $\boldsymbol{d} \in D$ to be clustered into K clusters. The objective of K-means is to find a partition that minimizes the sum of squared error over K clusters. Let the mean of a cluster be μ_k and the points in the cluster be c_k, the objective function is defined as follows.

$$J(C) = \sum_{k=1}^{K} \sum_{d_i \in c_k} \|d_i - \mu_k\|^2 \tag{4.1}$$

K-means requires the number of clusters K to be predefined. The simple K-means clustering algorithm for K clusters is presented below.

Algorithm 1 Simple K-means Algorithm

```
Select $K$ points as the initial centroids.
repeat
    Assign each point to its closest centroid.
    Recompute the centroid of each cluster.
until the centroids no longer change.
```

4.4.1.2 Hierarchical Clustering

The average linkage hierarchical clustering (UPGMA) is used in this project. Hierarchical clustering is a clustering method which produces a series of partitions of data points [?]. The objects are initially assigned to their own cluster and then the pairs of clusters are continually merged until the dendrogram is formed.

Agglomerative and divisive approaches are two approaches to implement hierarchical clustering [?]. In clustering literature, the agglomerative approach is more common than the divisive approach. While the divisive approach builds a hierarchy from top-down, the agglomerative approach builds it from bottom-up. The agglomerative hierarchical clustering algorithm works as follows.

Algorithm 2 Agglomerative Hierarchical Clustering Algorithm

```
Initialize each document $d_1, ..., d_N$ as its own cluster $C_1, ..., C_K$.
Compute the similarity or distance measure between all pairs of clusters and store them as a similarity matrix whose $ij^{th}$ entry represents the similarity between $C_i$ and $C_j$.
Merge the most similar pair of clusters into one cluster.
Recompute the matrix of pairwise similarities for the new clusters.
Repeat steps 3 and 4 until only a single cluster remains.
```

Variations of agglomerative approach may employ different a linkage criterion such as single linkage, complete linkage, average linkage, and ward [? ?]. The comparison between different linkage criterion algorithms shows that the average linkage or unweighted pair group method with arithmetic mean (UPGMA) is the most accurate one [?]. UPGMA defines the linkage of the clusters as follows.

$$d(C_i, C_j) = \frac{\sum_{\substack{d_i \in C_i \\ d_j \in C_j}} d(d_i, d_j)}{|C_i||C_j|} \tag{4.2}$$

The number of clusters of hierarchical clustering can be determined by a threshold or a cut-off placed on the similarity measure used in order to obtain the final clusters [**?**]. There are various ways to determine thresholds, for example, the predetermined number of clusters and distance partition.

4.4.2 Topic Modeling

In this project, we use a standard topic modeling method to find topic words from the document collection, which is latent dirichlet allocation (LDA).

4.4.2.1 Latent Dirichlet Allocation

Latent dirichlet allocation (LDA) is a probabilistic generative model to model texts and identify the hidden thematic structure in a collection of documents [**?**]. It is central to topic modeling and often used as the benchmark in topic modeling studies.

LDA makes three assumptions. The first assumption that LDA makes is the bag-of-words assumption. Another assumption is that the order of documents does not matter. The third assumption is that the number of topics is assumed to be known and fixed.

LDA assumes the generative process for a document collection. The generation process for a corpus D is as follows:

1. For topics $k \in \{1, ..., K\}$
   (a) $\phi_k \sim Dirichlet(\beta)$
2. For documents $d \in \{1, ..., D\}$
   (a) $\theta_d \sim Dirichlet(\alpha)$
   (b) For words $w_i \in d$
      i. $z_i \sim Multinomial(\theta_d)$
      ii. $w_i \sim Multinomial(\phi_{z_i})$

The dependencies between various variables in this generative process can be formally described with the following notations. ϕ is the matrix of topic distributions over each word, drawn independently from the Dirichlet (β) prior. θ is the matrix of topic distributions over each document, drawn independently from the Dirichlet (α) prior. z denotes the topic responsible for generating the observed word w, drawn from the θ distribution for that document. w is the observed word, drawn from the topic distribution ϕ_z corresponding to z. Therefore, the joint probability of hidden and observed variables can be described as follows.

$$p(w, z, \theta, \phi|\alpha, \beta) = p(\phi|\beta)p(\theta|\alpha)p(z|\theta)p(w|\phi_z) \tag{4.3}$$

Various algorithms have been used to estimate the hidden variables (ϕ and θ) which depend on α and β. While α is a hyper-parameter, we have to approximate β using a sampling method. The commonly used sampling algorithm is Gibbs sampling [**?**]. However, Gibbs sampling does not scale well with a large data set since it requires the entire corpus to be loaded into a memory and we have to rerun the algorithm when new data are added. Next, an online variational Bayes (online VB) algorithm for LDA is developed solve this problem and it is proved to scale with an arbitrarily large corpus [**?**]. The online VB works as follows.

Algorithm 3 Online VB

```
Randomly initialize $\lambda$, the variational parameter for $\beta$
Set the step-size schedule $\rho_t$
for batches $t = 0$ to $\infty$ do
    Initialize $\gamma_{tk} = 1$
    repeat
        Compute $\phi_{twk} \propto exp\{E_q[log\theta_{tk}] + E_q[log\beta_{kw}]\}$
        Compute $\gamma_{tk} = \alpha + \sum_w \phi_{twk} n_{tw}$
    until the local variational parameters $(\phi_{twk}, \gamma_{tk})$ converge
    for topics $k \in \{1, ..., K\}$ do
        Compute $\bar{\lambda}_{kw} = \eta + D\eta_{tw}\phi_{twk}$
    Set $\lambda = (1 - \rho_t)\lambda + \rho_t\bar{\lambda}$
```

In Algorithm 3, we sample a chunk of documents uniformly from the corpus for each batch t. Then we compute the local variational parameters ϕ and γ for the batch. Lastly, we compute the global variational parameters $\bar{\lambda}$ and merge them with the overall variational parameter λ, the variational posterior over the topic distributions β.

4.4.3 Network Analysis

Our approach mainly consists of four operations: document network construction, network preprocessing, community detection, and topic word extraction.

4.4.3.1 Document Network

The proposed model will be applied to a document network. The network is a weighted undirected graph. The graph is constructed out of a document collection in which nodes represent the TF-IDF document vectors and weighted edges represent the cosine similarity between two vectors.

The weighted edges are represented by cosine similarity between two documents. Cosine similarity between d_i and d_j is defined as:

$$cos(\boldsymbol{d_i}, \boldsymbol{d_j}) = \frac{\sum_{k=1}^{n} w_{ik} w_{jk}}{|\boldsymbol{d_i}||\boldsymbol{d_j}|}$$

where $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$ are the weighted term vectors and $|\boldsymbol{d_i}|$ is the length of the document vector $\boldsymbol{d_i}$ and $|\boldsymbol{d_j}|$ is the length of the document vector $\boldsymbol{d_j}$.

4.4.3.2 Network Preprocessing - The Selection of Optimal Threshold

Before applying the community detection method to the network, the network has to be preprocessed. The reason is because the network starts off as a complete network in which every node connects to each other. The problem is that this complete network does not allow the community detection to function optimally.

The common practice in network analysis is to prune the graph by removing irrelevant nodes from the graph. Since the edges of our graph are weighted by cosine similarity, one method is to use a cut-off threshold applied to the weighted edges in order to keep only important edges. This removes dissimilar documents measured by the similarity measure from the network. In our case, the removed documents are considered as uncategorized or having its own topic.

The selection of the optimal threshold is based on the empirical analysis. The choice of the threshold is based on four conditions. Firstly, the optimal threshold is the one that initiates the breakdown of the largest connected component. The largest connected component is the largest subgraph in which any two nodes are connected to each other by edges and they are not connected to nodes from other subgraphs. Secondly, the average degree should be small. The average degree is the average number of edges that a node has. Thirdly, the modularity value should be higher than 0.3 [**?**]. Lastly, the number of communities should be between 50 to 250 which is not too high or too low for the size of the document collection. Then we will apply the community detection on the reduced network.

4.4.3.3 Louvain Modularity for Community Detection

The Louvain method is the modularity-based approach for community detection which aims to find communities by optimizing modularity [**?**]. The algorithm is proven to be well-suited for a large network compared to the original Girvan and Newman's method and some of the other modularity optimization. The algorithm is divided into two phases. All nodes N start as its own community. During phase 1, for each node i we consider the neighbors j of i and calculate the gain of modularity by moving i from its community and to the community of j. The node i is placed in the community that results in the largest increase of modularity. This phase is repeated until the algorithm reaches the maximum modularity. Phase 2 is about the creation of the reduced network. The newly formed communities are considered as nodes in a reduced network. The weighted edges between two communities are the sum of the edge weights between the lower-level nodes in each community. The edges within each community are self-loops in the reduced network.

The common definition of modularity used in the original paper on the Louvain method, which is equivalent to the Girvan and Newman's one, is defined in Equation 4.4.

$$Q = \frac{1}{2m} \sum_{i,j=1}^{N} \left[A_{i,j} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{4.4}$$

Consider a network with N nodes, A_{ij} represents the edge weight between i and j. $k_i = \sum_j A_{ij}$ is the sum of the edge weights attached to node i. c_i is the community to which node i is assigned, the δ-function $\delta(c_i, c_j)$ is 1 if $c_i = c_j$ and 0 otherwise. Lastly, $m = \frac{1}{2} \sum_{i,j}^{N} A_{ij}$

4.4.3.4 Topic Word Extraction

Similar to topic modeling, we need to determine the topic words that describes a cluster of documents. Our approach to extract the set of topic words is to count the top TF-IDF word co-occurrences of a topic.

Algorithm 4 shows the process of our topic word extraction. For a given

Algorithm 4 Topic Word Extraction Algorithm for a Given Topic

```
repeat
    Retrieve $w_1, ..., w_n$ words from a word list of a document sorted by TF-IDF values
    Store them in a list
until Process all documents within a topic
Count word co-occurrences within the list
Select top-n highest frequent words from the word co-occurrences
```

topic, we retrieve top-n words sorted by TF-IDF values from each document within a topic. Then, we count the word co-occurrences of the retrieved word set and select the top-n most common words to represent the given topic. The topic words from this approach are more coarse-grain than simply retrieving the words sorted by TF-IDF values only.

4.5 Tasks

The project is divided into two main tasks. In the first task, the Guardian's dataset will be clustered by using three different methods, which are K-means, hierarchical clustering, and Louvain modularity. In the second task, we extract the topic words from the clusters of Louvain modularity and compare them with the ones from LDA. The model performances of both tasks will be evaluated by qualitative and quantitative approaches.

Chapter 5

Task 1: Cluster Quality Comparison

The first task is designed to evaluate the clustering performance of our approach compared to the other methods, namely K-means and hierarchical clustering. The general notion of good clustering is that the similar documents are put in the same cluster and the dissimilar ones are separated to the other clusters.

5.1 Experiment Setup

5.1.1 Dataset

The dataset used in this task contains the documents from Section 4.3.

5.1.1.1 Document Preprocessing

To prepare documents for this experiment, the documents are parsed out of JSON objects. The words are stemmed by Porter Stemmer and tokenized. The stopwords, numeric words and non-latin characters are removed. Finally, we use the part-of-speech tagging to identify and extract only nouns and proper nouns from the word tokens.

5.1.1.2 Document Representation

In this task, a document is represented by the TF-IDF (term frequency–inverse document frequency) weighting scheme is the most common metric for the vector space model. Its intuition is that an infrequent term that appears in a few documents is a better discriminator than a frequent term found in many documents. According to this rule, the infrequent term will have a low IDF while the frequent term will have higher IDF value. The TF-IDF index is the product of two factors: the term frequency and the logarithm of the inverse document frequency. TF of a term t_j in the document d_i is the number of occurrences of t_j in d_i, divided by the total number of all terms in the document:

$$TF(d_i, t_i) = \frac{n_{ij}}{\sum_{k=1}^{m} n_{ik}} \tag{5.1}$$

where n_{ij} is the number of occurrences of the term t_j in the document d_i.

IDF of the term t_j can be computed as:

$$IDF(t_j) = log(\frac{N}{N_j}) \tag{5.2}$$

where N_j is the number of documents that contains the term t_j.

5.1.2 Tools and Settings

All of our experiment code is in Python. Three approaches are tested in this comparison, as in the following list:

- Louvain Modularity

- K-means
- Hierarchical Clustering

The configuration of Louvain modularity is the same as mentioned in Section 4.4.3.3. There is no parameter to be set for Louvain modularity. The number of clusters is automatically determined by its algorithm. The implementation of Louvain modularity is from the open source package, igraph[1].

For K-means, we do not set the parameters for K-means apart from the number of topics. However, K-means scores will be based on average scores of 5 runs. This is because the random initial points influence the scores. K-means is performed using the scikit-learn[2] package.

For hierarchical clustering, we use the average linkage clustering method (UPGMA) and the cosine distance metric. The implementation of hierarchical clustering is from the scipy[3] package.

5.1.2.1 The Choice of the Number of Clusters k

One major obstacle with the other methods in this study is that they take in the number of clusters k as an input to the algorithm while Louvain modularity does not take. This is problematic when the actual k value is not known. Choosing the right number of clusters k is the controversial issue in clustering [**?**]. Nevertheless, the comparison between each clustering method usually uses the same number of clusters. To keep our experiment simple, our experiment will be a comparison of clustering performance based on the number of clusters returned from of Louvain modularity. This can still demonstrate the performance of Louvain modularity.

[1] http://igraph.org

[2] http://scikit-learn.org

[3] https://www.scipy.org

Edge Threshold	Nodes	Edges	LCC	Avg. Degree	Maximum Modularity	No. of Communities
0.0	9095	41273155	9095	9094	-	-
0.1	9070	641316	9070	141.414	0.592	14
0.2	**8213**	**132243**	**7950**	**32.203**	**0.798**	**139**
0.3	6565	43866	5269	13.363	0.892	437
0.4	4854	17436	1933	7.184	0.933	675
0.5	3170	6951	243	4.385	0.953	669

Table 5.1: The Optimal Threshold of Weighted Edges

5.2 Results

5.2.1 The Optimal Threshold of Weighted Edges

Table 5.1 summarizes the network statistics for various cut-off thresholds of the weighted edges. The numbers of communities and the modularity values are the results from Louvain modularity applied to the reduced networks. According to the conditions in Section 4.4.3.2, the threshold of 0.2 is selected as the best threshold for this network. Considering the cut-off thresholds from 0.1 to 0.2. Specifically, 1,120 nodes are removed from the largest connected components (LCC). The average degree reduces from 141 to 32. The modularity value is 0.798 which is higher than the usual value. The number of communities which is 139 is also reasonable. These evidences indicate that the nodes are more separated. As a result, the remaining 8,213 documents are categorized into 139 communities.

5.3 Evaluation for Cluster Quality

After we have the result from the previous section, K-means and hierarchical clustering are used to find the clusters in the remaining documents for comparison. The number of clusters for all methods is 139 clusters.

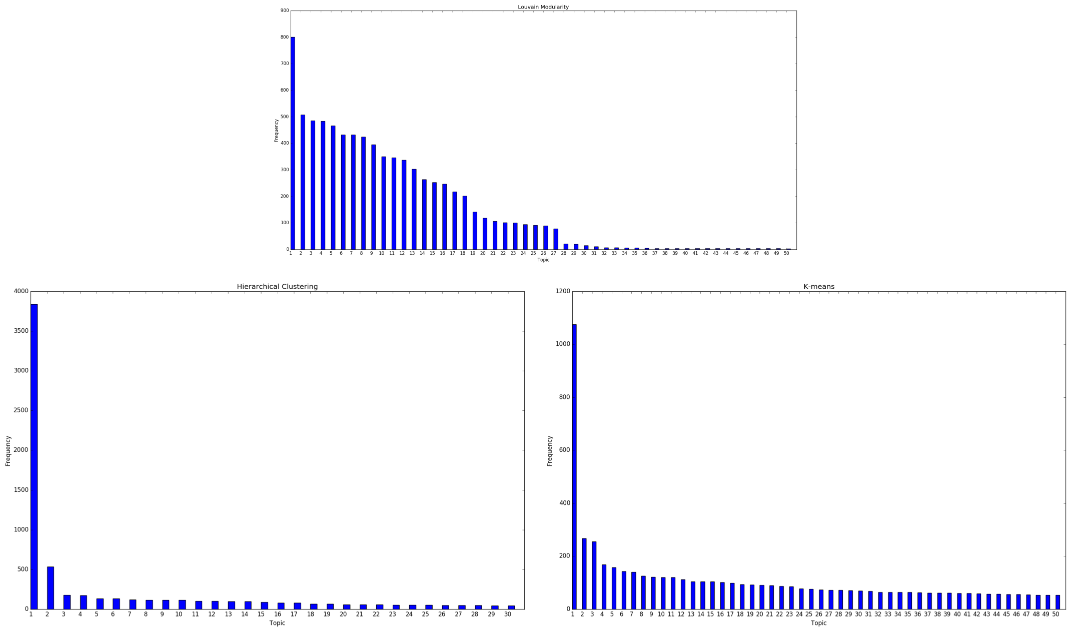

Figure 5.1: Topic distributions of each method. From Left to Right: Louvain Modularity, Hirarchical Clustering, K-means

5.3.1 Qualitative Evaluation

Figure 5.1 shows the topic distributions of Louvain modularity and the other methods. The topics are ordered by their topic sizes. For clear visibility, the figures only display 50 topics.

The topic distribution of Louvain modularity exhibits the skewed shape indicating the imbalance of dataset. In contrast, the distributions of the other methods are uniformly distributed with one large topic. 90% of documents are allocated to the top-30 topics for Louvain modularity while the other methods distribute data evenly to many topics.

The probabilistic distributions of topic sizes of each method are plotted in Figure 5.2. For other methods, they seem to produce many small topic sizes. Most of the topic sizes of K-means and hierarchical clustering are approximately ranged from 100 to 200 documents. This confirms that the

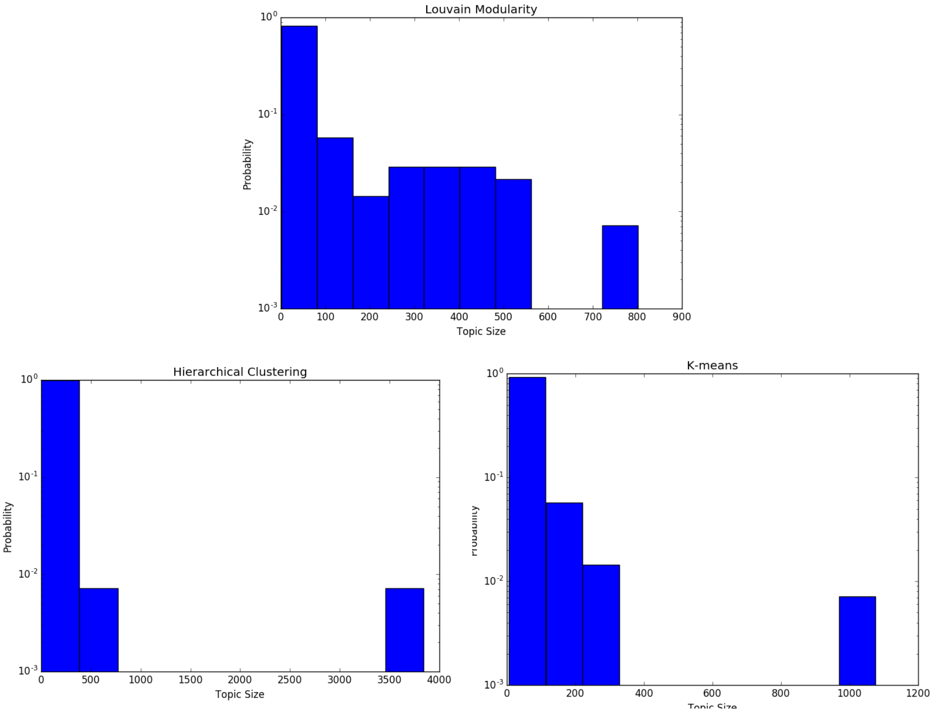

Figure 5.2: The probability distributions of topic sizes of each method. From Left to Right: Louvain Modularity, Hirarchical Clustering, K-means

documents are distributed uniformly across different small topics. Whereas, nearly 30 percent of the topics from Louvain modularity contains between 200 and 500 documents.

To get a clearer insight, we apply the topic word extraction in Section 4.4.3.4 on the topics of each method to retrieve their topic words. Then, we pick the topics that have the words, "footbal" and "goal", in order to see how many football-related topics are there. As can be seen in Table 5.2, there are 5 topics, 3 topics, and 2 topics related to football for K-means, hierarchical clustering, and Louvain modularity respectively.

Practically, all football-related documents should be assigned to the same group. However, K-means separates them into 5 different topics and hierarchical clustering allocates them into 3 topics. On the contrary, Louvain mod-

Method	Topic	Size	Words
K-means	3	256	season, game, leagu, player, tottenham, goal, chelsea, manag, match, refere
	55	50	tico, game, goal, madrid, minut, atl, champion, simeon, player, team
	95	29	sunderland, season, game, goal, chelsea, player, leagu, allardyc, everton, team
	96	29	ranger, peopl, commun, club, wildlif, footbal, ibrox, fund, park, manag
	128	15	club, footbal, fan, leagu, player, support, season, game, stadium, year
Hierarchical Clustering	2	534	game, player, season, club, team, leagu, goal, year, minut, champion
	3	179	season, leagu, club, player, footbal, leicest, game, team, manag, citi
	53	22	season, team, leagu, club, citi, manchest, game, player, premier, goal
Louvain Modularity	1	801	season, player, game, club, leagu, team, footbal, goal, manag, manchest
	42	4	game, aleagu, goal, sydney, wander, isaia, season, adelaid, footbal, deduct

Table 5.2: The topics containing ”footbal” and ”goal”

ularity has only 2 football-related topics. Although this result is imperfect, there are only 4 documents that are separated into the other topic. Nonetheless, these results show that Louvain modularity achieves significantly fewer errors than the other methods. Consequently, its quality of topics is also better than the other methods.

5.3.2 Quantitative Evaluation

We use two metrics to measure the cluster quality: intra-cluster cosine similarity and inter-cluster cosine similarity. These metrics are internal metrics used when the ground-truth is not known. The detailed explanations of our metrics are as follows:

Intra-cluster cosine similarity measures the level of cohesion within a cluster. The high intra-cluster cosine similarity implies that the cluster

Method	Intra-Cluster Similarity	Inter-Cluster Similarity
K-means	0.243	0.114
Hierarchical Clustering	0.480	0.101
Modularity	**0.525**	**0.051**

Table 5.3: The Clustering Quality Results

consists of similar documents. In our experiment, this is calculated by the arithmetic mean of intra-cluster cosine similarity of all clusters. The intra-cluster cosine similarity of a given cluster is the cosine similarity of TF-IDF document vectors within the same cluster. The intra-cluster cosine similarity of a given cluster is defined as in equation 5.3:

$$\frac{1}{|C|^2} \sum_{\substack{d_i \in C \\ d_j \in C}} cos(d_i, d_j) \tag{5.3}$$

where C is a given cluster, $|C|$ is the number of vectors in the cluster, and d is a document vector.

Inter-cluster cosine similarity measures the level of separation among clusters. The low inter-cluster cosine similarity indicates that a cluster is significantly different from the others. In our experiment, this is calculated by the arithmetic mean of cosine similarity between each cluster. We use the centroids of each cluster to plot an array of confusion matrices. The centroids are calculated by the aggregated TF-IDF vectors in each cluster. The inter-cluster cosine similarity of a cluster C_i and the other clusters C_j is defined as in equation 5.4:

$$\frac{1}{|S|} \sum_{\substack{C_i \in S \\ C_j \in S}} cos(C_i, C_j) \tag{5.4}$$

where C is a given cluster and $|S|$ is the number of clusters in a set of clusters.

Table 5.3 displays the evaluation results of each method. Louvain modularity's clusters give the best performance because it does not only group more similar documents inside the same clusters but also separate less similar documents into the other clusters. K-means has the poorest performance of three methods and even poorer than hierarchical clustering.

Chapter 6

Task 2: Topic Word Quality Comparison

The second set of experiments is designed to evaluate the topic words of our approach compared to the standard method, namely LDA. The general notion of the good topic words can be simply described as the best semantic labels for a given topic.

6.1 Experiment Setup

6.1.1 Dataset

The dataset used in this experiment is from Section 4.3.

6.1.1.1 Document Preprocessing

Similar to the previous task, the documents are parsed out of JSON objects. The common preprocessing tasks are applied including tokenization, stop word removal and stemming. Numeric words and non-latin characters are cleaned. Finally, the part-of-speech tagging is used to extract only nouns and proper nouns from the word tokens.

6.1.2 Tools and Settings

In this experiment, we use only two methods which are LDA and Louvain modularity.

- Louvain Modularity
- LDA

The setting of Louvain modularity is the same as the previous task. The implementation of latent dirichlet allocation (LDA) is based on Vowpal Wabbit[1] and it is executed under the gensim[2] wrapper. The model settings will be determined by the perplexity value.

6.2 Results

6.2.1 Louvain Modularity Results

For Louvain modularity, we borrow the result from the clustering section. Therefore, the number of topics is determined by the same approach. 8,213 documents are assigned to 139 topics. Then, topic words are retrieved from the topics using the topic word extraction method.

6.2.1.1 Topic Words from Louvain Modularity

Table 6.1 shows the top-10 words of top-30 topics retrieved from Louvain modularity. The topics are sorted by their document frequencies. These 30 topics contain approximately 90% of documents in the original dataset. With the document frequencies and the topic words presented in the table, we can

[1] https://github.com/JohnLangford/vowpal_wabbit

[2] https://radimrehurek.com/gensim

quickly capture the main themes in the document collection by skimming through the table.

As can be seen, the topic words can effectively imply the specific semantics of each topic. For instance, in Topic 1, we can see that the words "season", "player", "league", and "football" all indicate that the topic is about sports and football news. In Topic 2, the documents under this topic are about films and movies. In Topic 3, most of the words are about children education and refugee education. Lastly, in Topic 4, all words are closely associated with the climate change and environmental issues.

These topic words are also highly related to its own topic and we can semantically differentiate a single topic from the other topics. From the table, the two most likely similar topics seem to be Topic 3 and Topic 18. However, there is a slight difference if we investigate more closely. Topic 3 is more about children's education while Topic 18 is more about the higher level education.

The level of granularity depends on the size of the topic. The higher the number of documents in a topic is, the more general the topic words are. For instance, Topic 1, which is related to sports and football news, have general words such as "season", "player", and "game" to represent the topic. On the other hand, Topic 20 which is about cricket news contains a domain-specific vocabulary such as "wicket", "inning", and "bowler".

When the topics have a low number of documents, some topic words are too specific and difficult to be interpreted. For example, Topic 28, which is about puzzles, only has 21 documents. It contains the unique words such as "gaffer", "gaffa", "gaff". Our topic word extraction algorithm has the difficulty to distinguish between common words and rare words if a given topic has only a few number of documents.

Topic	Size	Topic Words
1	801	season, player, game, club, leagu, team, footbal, goal, manag, manchest
2	508	film, women, movi, peopl, men, girl, director, cann, woman, year
3	486	govern, children, school, refuge, peopl, educ, parent, countri, teacher, year
4	484	climat, citi, energi, compani, govern, chang, car, year, water, peopl
5	467	music, song, album, band, year, festiv, peopl, audienc, record, artist
6	432	year, market, bank, price, hous, peopl, home, compani, properti, custom
7	432	eu, brexit, referendum, britain, vote, campaign, europ, cameron, uk, johnson
8	424	health, peopl, care, servic, patient, work, govern, year, nh, hospit
9	395	labor, govern, tax, elect, turnbul, minist, campaign, coalit, polici, budget
10	350	parti, elect, labour, khan, leader, mayor, corbyn, tori, seat, mp
11	346	art, game, artist, museum, galleri, peopl, work, war, exhibit, world
12	337	busi, compani, media, competit, entri, award, categori, judg, panel, shortlist
13	303	trump, clinton, campaign, state, candid, presid, parti, sander, nomine, nomin
14	264	polic, offic, court, investig, case, forc, man, justic, report, law
15	253	book, stori, site, world, youv, bookshop, review, charact, novel, children
16	247	prison, drug, court, peopl, justic, year, case, death, law, use
17	218	bbc, broadcast, govern, corpor, tv, paper, whittingdal, licenc, charter, channel
18	202	student, univers, peopl, educ, govern, year, research, compani, scienc, school
19	141	race, hors, winner, driver, stake, trainer, derbi, hamilton, colt, time
20	118	cricket, test, england, ball, wicket, inning, lanka, bowler, match, run
21	106	pension, busi, bh, fund, committe, compani, govern, mp, acquisit, retail
22	101	stori, contribut, guardianwit, review, presid, video, photo, creation, button, pictur
23	100	garden, plant, speci, flower, wildlif, year, anim, photograph, chelsea, water
24	94	church, murray, match, djokov, tournament, tenni, player, time, garro, game
25	91	olymp, athlet, game, rio, championship, golf, sport, world, ioc, medal
26	89	food, egg, recip, oil, dish, minut, salt, cook, butter, bowl
27	78	flight, plane, passeng, egyptair, aircraft, ms804, air, airlin, bomb, airport
28	21	puzzl, gaffer, gaffa, gaff, gaelic, week, gaffigan, number, gaetano, gag
29	20	disciplin, season, sub, form, aug, kickoff, d, scorer, knee, r3
30	15	wine, vineyard, grape, sauvignon, daveydaibach, savouri, bottl, abv, blanc, loir
Total	7923	

Table 6.1: The top-30 topics and top-10 topic words

6.2.1.2 Visualization

Figure 6.1 shows the visualization of a network of top-30 topics. Each topic is labeled with the top-5 topic words retrieved from the topic word extraction. Each edge represents the word co-occurrence counts of topic words from each topic. The nodes of the network are manually positioned for better visibility.

This visualization demonstrates the capability of network analysis. It

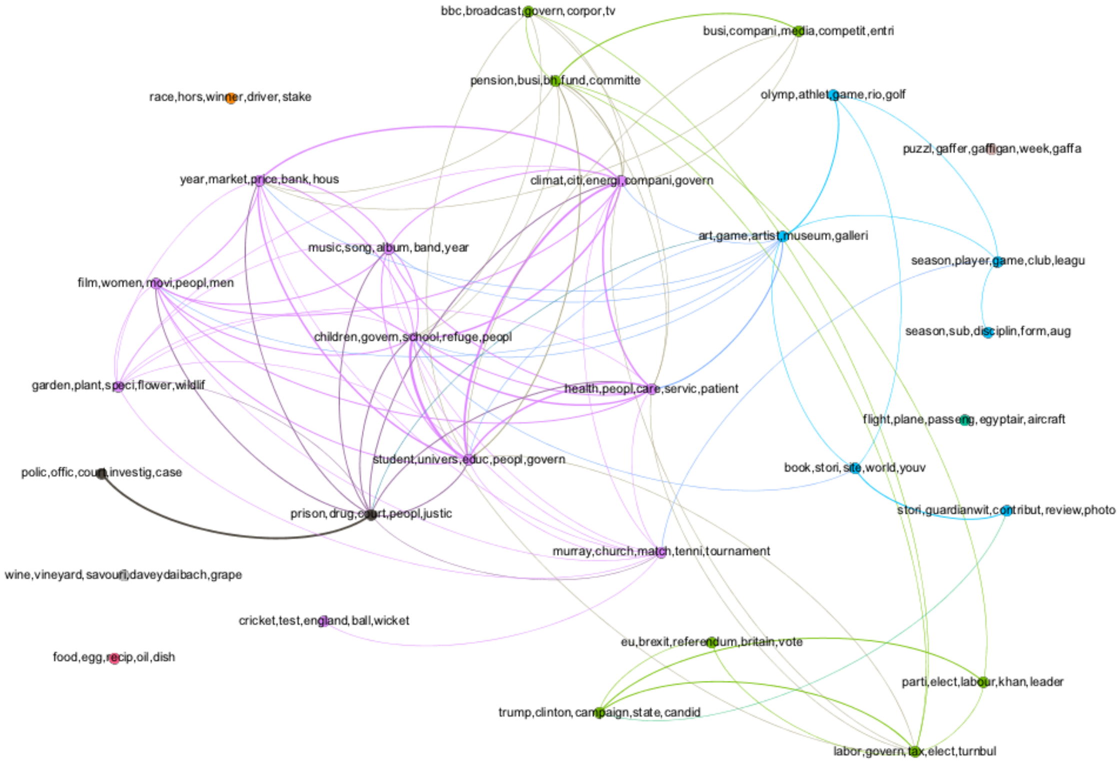
bbc,broadcast,govern,corpor,tv
busi,compani,media,competit,entri
pension,busi,bh,fund,committe
race,hors,winner,driver,stake
olymp,athlet,game,rio,golf
puzzl,gaffer,gaffigan,week,gaffa
year,market,price,bank,hous
climat,citi,energi,compani,govern
art,game,artist,museum,galleri
music,song,album,band,year
season,player,game,club,leagu
film,women,movi,peopl,men
season,sub,disciplin,form,aug
children,govern,school,refuge,peopl
garden,plant,speci,flower,wildlif
health,peopl,care,servic,patient
flight,plane,passeng,egyptair,aircraft
book,stori,site,world,youv
polic,offic,court,investig,case
student,univers,educ,peopl,govern
stori,guardianwit,contribut,review,photo
prison,drug,court,peopl,justic
murray,church,match,tenni,tournament
wine,vineyard,savouri,daveydaibach,grape
cricket,test,england,ball,wicket
eu,brexit,referendum,britain,vote
food,egg,recip,oil,dish
parti,elect,labour,khan,leader
trump,clinton,campaign,state,candid
labor,govern,tax,elect,turnbul

Figure 6.1: Visualization of Topic Words

shows the relationships among topics in the document collection. Without knowing anything about the document collection before, we can look at the visualization and understand the major stories of the document collection in May 2016. From Figure 6.1, the bottom right corner of the network contains a group of political topics. There are four related topics which are "BREXIT", "US Election", "UK Election", "Australia Election". The lifestyle and general news are displayed on the left side of the network. For instance, the general news includes "health", "music", and "climate". The right side of the network has contents related to sports, for example, "Olympics" and "football". Moreover, we can move from one topic to another relevant topic by following the edge. For instance, we can see that the edge connects the book topic to the children's topic.

Topics (k)	10	20	30	40	50
Perplexity	5463.527	5281.806	5228.749	**5193.834**	5333.831

Table 6.2: Comparison of perplexity values between the numbers of topics

6.2.2 LDA Results

In this task, we find the best model of LDA. The search for the best LDA model starts with the number of topics (k). The commonly used technique to determine the number of topics is perplexity. Perplexity is the standard way for model selection in the field of topic modeling. The perplexity on a held-out dataset is used to measure the overall fitness of the natural language processing model on the given dataset. The lower the perplexity, the better the generalization of the model. The perplexity is monotonically decreasing in the likelihood of a held-out dataset. It has equivalent to the exponential of the negative average per-word log-likelihood. The equation is formally defined in equation 6.1 [**?**].

$$perplexity(D_{test}) = exp\Big\{ - \frac{\sum_{d=1}^{M} log\ p(w_d)}{\sum_{d=1}^{M} N_d} \Big\} \tag{6.1}$$

The dirichlet priors, α and β, are set to $1/k$ to allow the optimal number to be determined in a data-driven fashion. Moreover, we set the number of topics (k) as 10,20,30,40,50 for the search of the best number. The dataset is divided into a training set and a test set. We use 10-fold cross validation and compute the average perplexity for each number.

6.2.2.1 The Optimal Number of Topics for LDA

Table 6.2 shows the perplexity of each number of topics. For each number of topics, their perplexity values are averaged from 10-fold cross validation on a held-out dataset with 50 iterations on each run. According to these results, the lowest value is at 40 topics, meaning that this number of topics achieves better fitness of the dataset. Thus, we will use this model setup to infer the topics from the dataset.

6.3 Evaluation of Topic Words

6.3.1 Qualitative Evaluation

As shown in Table 6.3, the first column contains the topic labels defined manually based on the content on the right columns. The second column consists of the topic words from LDA and the topic words from the third column are from our method. To create this table, we first pick the topics from LDA results and find the topics that match the former topics from the results of our approach. The topic words are the top-10 words sampled from each method. The words in bold are common words between two approaches. We did not rank the results of LDA because LDA does not have the default ranking capability.

The topic words from our approach are generally similar to the ones from LDA and both sets of words are notably related to the topic labels. For instance, both methods have "music", "song", and "band" under the music topic. In contrast, some topic words from both approaches are not highly related to their topic labels. For example, "food" and "egg" are not semantically related to the health topic, whereas "women", "people", and "men", which are the topic words from our approach, are too general to represent the film topic. Despite the mentioned flaw, the topic words from our approach can semantically represent each topic in general.

Topic	LDA	Louvain Modularity
Sport	**game**, **season**, **player**, **club**, team, leagu, **footbal**, citi, **manag**, fan	**season**, **player**, **game**, **club**, **leagu**, **team**, **footbal**, goal, **manag**, manchest
Film	**film**, **movi**, actor, festiv, **director**, charact, star, seri, play, product	**film**, women, **movi**, peopl, men, girl, **director**, cann, woman, year
Novel	**book**, art, work, **children**, artist, **novel**, museum, writer, author, world	**book**, stori, site, world, youv, bookshop, review, charact, **novel**, **children**
Environment	**water**, **climat**, pollut, **chang**, reef, environ, barrier, river, scientist, bha	**climat**, citi, energi, compani, govern, **chang**, car, year, **water**, peopl
Music	**music**, **song**, **band**, **album**, **record**, **artist**, track, perform, video, ticket	**music**, **song**, **album**, **band**, year, festiv, peopl, audienc, **record**, **artist**
Business	**year**, **peopl**, **compani**, busi, servic, time, govern, work, **market**, job	**year**, **market**, bank, price, hous, **peopl**, home, **compani**, properti, custom
US Politics	**trump**, **clinton**, **parti**, **presid**, **state**, **campaign**, **candid**, **sander**, elect, obama	**trump**, **clinton**, **campaign**, **state**, **candid**, **presid**, **parti**, sander, nomine, nomin
Health	food, **health**, **patient**, doctor, **hospit**, **care**, cancer, egg, **nh**, restaur	**health**, peopl, **care**, servic, **patient**, work, govern, year, **nh**, **hospit**
AUS Politics	**labor**, **govern**, women, australia, **elect**, parti, **polici**, **minist**, **campaign**, **turnbul**	**labor**, **govern**, tax, **elect**, **turnbul**, **minist**, **campaign**, coalit, **polici**, budget
UK Politics	**eu**, parti, **campaign**, **vote**, labour, elect, minist, **britain**, **cameron**, leader	**eu**, brexit, referendum, **britain**, **vote**, **campaign**, europ, **cameron**, uk, johnson

Table 6.3: Comparison of Topic Words from Each Method

6.3.2 Quantitative Evaluation

This evaluation aims to measure the semantic quality of topic words using the automatic topic coherence evaluation metrics as follows: C_V and UMass. The intuition behind this evaluation is that coherent topic words should frequently appear together across the corpus.

CV is a recent work on the automatic evaluation of topic coherence [**?**]. The C_V metric is an extrinsic metric which uses the external corpus rather than the corpus that generates the topics. We use the evaluation framework called Palmetto[3] with the Wikipedia corpus as a data source for C_V which is the same corpus from the original paper [**?**].

The C_V metric consists of four parts that work together which are segmentation, probability estimation, confirmation measure, and aggregator. The standard notation and setup of this measure is in equation 6.2 as follows:

$$score_{C_v} = (S_{set}^{one}, P_{sw(110)}, \tilde{m}_{cos(nlr,1)}, \sigma_\alpha) \tag{6.2}$$

The segmentation (S_{set}^{one}) computes co-occurrence counts for the given words using a sliding window approach and the window size is 110 in this case. The probability estimation ($P_{sw(110)}$) estimates word probabilities in which the word occurs divided by the the number of sliding windows from the segmentation. The confirmation measure ($\tilde{m}_{cos(nlr,1)}$) computes normalized Point-wise Mutual Information (NPMI) of every word to every other word which results in a set of word vectors. This leads to the calculation of cosine similarity between every word vector to the sum of all word vectors. Under the aggregator (σ_α), the aggregated topic coherence score is calculated as the arithmetic mean of these similarities.

UMass is the commonly used topic coherence measure [**?**]. The gensim[4] implementation of UMass is used for this experiment. The UMass metric is an intrinsic metric which uses the original corpus used to train the topic models for evaluation. It attempts to confirm that the models have learned data from the trained corpus. The metric is based on document co-occurrence counts of words. The UMass score of two words is computed with the equation 6.3 as follows:

$$score_{UMass}(w_i, w_j) = log\frac{D(w_i, w_j) + 1}{D(w_j)} \tag{6.3}$$

[3] https://github.com/AKSW/Palmetto

[4] https://radimrehurek.com/gensim

Top-N Words	Method	CV	UMass
N=5	LDA k=40	0.535	-8.691
	Louvain Modularity k=30	**0.537**	**-2.194**
	Louvain Modularty k=139	0.514	-3.519
N=10	LDA k=40	**0.473**	-10.055
	Louvain Modularity k=30	0.443	**-2.167**
	Louvain Modularty k=139	0.433	-4.037

Table 6.4: Evaluation of Topic coherence

$D(w_i, w_j)$ counts the number of documents containing words w_i and w_j while $D(w_j)$ counts the number of documents containing w_j. The aggregated topic coherence score is calculated as the sum of these scores.

From the Table 6.4, N indicates the top-n number of topic words from each topic that is used for evaluation. The k number indicates the top-k number of topics that is evaluated. Louvain modularity has two numbers of topics (k) because the majority of the documents are within 30 topics.

The C_V scores for each method is listed in Table 6.4. For top-5 words, both methods perform evenly when we select the top-30 topics from Louvain modularity for comparison. However, the result from Louvain modularity decreases if all topics are in evaluation. This effect is expected because the result includes minority topics which have many domain-specific words. For top-10 words, the C_V score from LDA is higher than the scores of Louvain modularity by 3%.

Interestingly, the UMass metric, which is based on the internal corpus, rate the topic words from our approach higher than LDA for both top-5 and top-10 words. This shows that our topic words are more coherent than the topic words from LDA, meaning that our approach retrieves more frequently co-occurred words from the internal corpus to represent each topic.

Chapter 7

Discussions

7.1 Summary of Experiments

The comparison of cluster quality gives an insight on how the network analysis approach can be better than the other clustering methods. Our approach can group similar documents into the same topic and separate dissimilar documents into the other topics. Moreover, our approach also generates coherent topic words for the discovered topics. The results from clusters positively influence the quality of the topic words. With the simple topic word extraction, the result is even better than LDA for the evaluation metric based on the internal corpus.

7.2 Intuitive Explanations of Our Performance

Overall, the quality of topics retrieved from our network analysis approach can be as good as the textual approach such as LDA. The topic coherence evaluation of the topic words from our approach indicates that the combinations of our topic words are rated as being coherent and understandable by humans. This proves that we can use the network analysis approach to reveal latent topics from the document collection.

The performance of our approach in document clustering is the key contributor to the quality of topic words. Based only on node similarity, Louvain modularity ensures that highly similar documents are grouped into the same topic and dissimilar documents are separated into other topics. This leads to good clusters of similar items. Despite using the simple topic extraction algorithm, we can achieve good coherent topic words when they are compared to the LDA results which are the standard model in topic modeling.

Our network-based approach can capture the topic structure from the imbalanced dataset. According to the results from Section 5.2, the topic distribution of our approach exhibits the skewed data distribution. In contrast, the clustering results from K-means and hierarchical clustering show relatively uniform distributions. As shown in the first task, if we use these clusters from the other methods to represent the document collection, we will have many of the similar semantic topics.

7.3 Limitations and Improvements

The first limitation of our approach is data reduction in the original dataset. The network preprocessing is crucial for community detection to work properly and improve the quality of the results. Although the preprocessing has removed some insignificant nodes from the network, the process has a trivial effect on our result. From the experiment, our approach is still able to infer the main topic structure from the original dataset compared to LDA. However, this preprocessing process can be improved to keep all nodes in a network.

The second limitation is that the retrieved number of topics cannot be adjusted. When the number of topics is given, we cannot force the community detection algorithm to find the topics at the given k. Under this approach, if a specific number of topics is needed to be extracted, the cut-off thresholds have to be adjusted until we get the number of topics that we need. Nonetheless, the choice of the number of topics depends on the level of granularity. If we need coarse-grained topics, we usually keep the low number of topics. If we need fine-grained topics, we set the number of topics higher.

The third limitation is that the topic words depend on the number of documents in a topic. The community detection algorithm produces many small size topics containing a few number of documents when it returns the large number of topics. The topic words inferred from these small topics are not generalized and very specific in a certain domain. Researchers should be aware of this caveat and they may choose to only consider the large topics if they want to infer the main topic structure of the document collection.

7.4 Implication of Our Work for Topic Modeling

As we have stated before, we do not aim to replace existing methods. We try to highlight the valuable, important information encoded in the network connectivity, which is profound and natural properties of documents, but have so far been overlooked. We demonstrate its capability in document clustering and topic modeling. As we can see, the combination of the two may potentially improve the topic modeling methods and solve some of the hard problems in this area.

The first implication of this approach is to automatically find the number of topics in an unsupervised manner. By using this approach, we do not have to go through the burden of finding the number of topics k for the LDA model. The number of topics obtained from the network-based analysis can be used as an initial guess for the optimal number of topics in the LDA model. From our experiment, our approach can discover approximately 30 major topics from the Guardian's document collection. Interestingly, the optimal number of topics according to the perplexity measure is also between 30 and

40 topics. Nevertheless, we need further studies to seamlessly combine these two approaches together.

The topic words produced from our approach can also be used to improve the initialization of LDA. For instance, the topic words of top-30 topics could be used as the initial estimation of the word composition of each topic before initializing the LDA model. However, we need to work out how to assign the word-topic probabilities to our topic word extraction method before integrating this into the LDA model.

7.5 Future Works

Firstly, we can further develop the proposed approach of this study. Network pruning is one of the areas where it can be improved. If the approach can prune the network while preserving important data and relationships among them, it can make our performance more accurate at marking an inference from a dataset. The common practice of network pruning is to apply a cut-off threshold on the weighted edges. This threshold method used in our model is proved to be able to extract important topics from the document corpus as good as LDA. However, it would be better if we have a more sophisticated method that can minimize the reduction in dataset. This method does not only complement the detection of topic structure but also benefit the other data mining tasks.

We also can update our community detection method. There are other community detection methods that can be used for clustering such as Infomap [**?**]. It is also interesting if we can investigate on all these methods in terms of its performance in detection of topic structure.

Next, our approach is different from other network-based research on topic modeling. Our approach works from document to topic words. In contrast, the other approach processes oppositely from topic words to documents. This approach usually creates a word co-occurrence network and carries out the classification of documents in the later stage. The relative performance of these approaches has not been investigated either. It would be interesting to compare these two approaches.

Lastly, we can take our research to the more sophisticated model which is the combination of network and LDA. As mentioned before, we can use the network analysis approach as the initialization step of LDA model in order to improve the performances of LDA in topic modeling tasks.

Chapter 8

Conclusion

8.1 Summary of Works and Achievements

Our experiments are designed to assess the network analysis approach on two different tasks which are clustering and topic word extraction. These two tasks constitute the detection of topic structure problem. Our dataset is from the document dataset retrieved from the open platform of the Guardian. Both qualitative and quantitative evaluations are employed to compare our approach with the commonly used methods in industry.

For the clustering task, we compare three different methods in order to benchmark the performance of our approach. The three methods are Louvain modularity, K-means, and hierarchical clustering. Since we do not have the precise ground-truth, we use internal metrics which are intra-cluster cosine similarity and inter-cluster cosine similarity to measure the performances of each method. From the clustering results, the network analysis approach outperforms other approaches. By examining the topic structure, the two other commonly used clustering methods, K-means and hierarchical clustering, tend to form clusters with a uniform distribution.

For the topic word extraction, we compared our approach with the standard textual method, latent dirichlet allocation (LDA) by using the recent topic coherence metrics, C_V and UMass. From the results, the topic word quality retrieved from our network-based approach can be as good as the textual approach. This is because our approach can discover good clusters which simplify the extraction of good coherent topic words.

The results of our experiments show a competitive quality of topics of the network analysis approach compared to the other standard methods. This confirms the value of the connectivity in a network. The implication of network analysis can be useful to textual approaches for the topic modeling problem.

8.2 Self-evaluation

From our experiments, the comparison of our network analysis approach to other methods is difficult. The reason is because of the other approaches, which are K-means, hierarchical clustering, and LDA, are either appropriate for clustering or discovering topic words. K-means and hierarchical clustering are not for extracting topic words. On the other hand, LDA is not used for clustering. Therefore, we have to separate the evaluations into two tasks in order to get the best out of the other methods.

8.3 Final Notes

The network analysis approach is proved to be effective in the detection of topic structure. Our main objective is achieved. It is hoped that we can improve our approach or combine the network analysis approach with LDA to improve its capability in topic modeling.

Appendix A

Source code

A.1 Initial setup

A.1.1 Github

`https://github.com/patorn/ucl_topic_structure_network_analysis`

A.1.2 Dataset

Requires the Guardians dataset. It can be found in data/ or download the new ones using the following scripts. The credential file (creds_guardian.txt) is needed. This can be obtained from `http://open-platform.theguardian.com/`

1. guardian_downloader.py is a script to download the Guardians web pages.

2. parser.py is a script to parse the web pages

A.1.3 Requirements

- jupyter python to open the ipython notebook
- Python 3.5.0
- igraph (Network)
- scipy
- scikit-learn
- gensim (LDA)
- Vowpal Wabbit (LDA)
- palmetto (CV Topic Coherence)

A.2 Manual

Follow the ipython notebook to complete the tasks.

task1.ipynb

Task 1 is to create a document network and find the network communities. We use the result of Louvain modularity to compare with the results from Hierarchical Clustering and K-means.

task2.ipynb

For Task 2, we find the topic words extracted from the network-based method and also find topics and topic words with LDA.

/results

This folder contains the results of various tasks including TF-IDF vectors, Network Preprocessing, LDA, and etc. that used in this project. In order to reproduce the same results, use the data in the folder.