Multi-Document Summarization of Public Health Literature

Niels Bantilan, Thomas Vo

1. Background

1.1 Motivation

GSumm is a literature review tool for journal articles in the public health sciences. The end goal of our project is to improve the PubMed search functionality by (a) topically organizing journal articles in the health sciences and (b) improving information retrieval for the end user through multi-document summarization

The set of machine learning tasks that we have defined to fulfill this goal is two fold. The first is to train an algorithm that automatically assigns topics to an article based on their text content, with as little dependency on metadata as possible, thereby allowing for the clustering of documents.

The second is to extract 'salient' sentences within these document clusters in order to provide the end user, in theory, with a quick survey of the document cluster's content based on a specific query. This would enable us to recommend articles that better suit their research needs, as well as to present only the most useful information to the user as the initial return to their query.

1.2 Rationale of Machine Learning Specifications

The main rationale behind the specification of these two tasks is that public health is a highly interdisciplinary field, and any given document is a mixture of many different topics. In addition, the terminology and jargon in the field is constantly evolving, and therefore we aim to build a system that does not necessarily depend on structured data (i.e. a prespecified hierarchy of topics that are created by experts), even though journal publications are typically well indexed with subject heading tags. Our reasoning behind this is that we hope to possibly extend our system to include less structured text examples, such as blogs, and news articles.

Furthermore, the sheer volume of peer reviewed journal material makes it increasingly challenging to perform literature reviews in the public health field. Therefore we hope to be able to more effectively cater articles based on a user's search that would potentially deliver a set of specific sentences or paragraphs pulling from multiple related documents.

Our machine learning task specification would allow us to create an automatic classification system for articles, which will help make informed recommendations about which articles are most relevant and should be returned when a user makes a query. This is the primary purpose of topic modeling: to give a meta-structure to the collection of articles, so as to help guide the search process.

For example, a researcher may want to do a literature review about "obesity in low income communities in the United States". After she enters this query into our system, she is returned five sentences that state the most important findings from all literature on the relevant topic and the

sources of the five topics. If she wants to read more, she can go directly to the source, which will be provided.

1.3 Project Scope

Our defined scope for this class is to specify the task, train an algorithm to learn from a set of observations, explore the initial results, and formulate a set of future directions moving forward. We are omitting the last part of the machine learning process (evaluation) for the current project because the document clustering and sentence extraction tasks that we have specified are unsupervised learning techniques. Therefore, we are not going to focus on how to assess the performance because we do not have explicit labels to obtain an error rate.

2. Methods

2.1 Creating the corpus

In gathering documents for our training corpus, we turned to the PubMed database, which is a comprehensive database of scientific articles that relate to the biomedical field and the health sciences. PubMed contains an open access subset of peer-reviewed scientific journals with freely available metadata and full text articles through their API.

A total of 1,000 full text articles were scraped from the Biomedical Central - Public Health journal for the current project. We wrote a script that queries their OAI service API in order to retrieve an XML response containing full text and metadata of each article. We used the BeautifulSoup package to retrieve the raw text from the XML file format and saved the documents to disk, each in a separate .txt file.

We then used the Gensim package (Řehůřek, 2014) to create a corpus, which we subsequently saved to disk. The Gensim package also creates a dictionary object, which consists of the set of all unique words in the entire corpus.

2.2 Pre-processing of Text

Before we could utilize the text data for machine learning, we needed to first transform each document's full text into a set of features. With our corpus, we treated each document as a bag of words (BOW). This means that each document is a collection of tokens, or instances of a word, regardless of order or structure.

We used routine NLP pre-processing techniques, namely stopword removal and removal of low frequency types, which reduces the dimensionality of the data by omitting uninformative words from each document. The remaining words were tokenized into features.

2.3 Feature Generation

2.3.1 Raw Term Frequency (tf)

One of the simplest ways to generate features from a text corpus is to count the word frequency of types (the set of all unique words in the corpus) to create a real-numbered vector representation of

each document. The resulting data structure would be a term-document matrix M, where each row 'i' is a document, and each column 'j' is a unique word. The value of the $M_{i,j}$ would be a real valued count of the word's raw frequency in a given document.

$$tf(t,d) = |\{d \in D : t \in d\}|$$

2.3.2 Term Frequency - Inverse Document Frequency (tfidf)

Tf-idf is a way to normalize the raw term frequency term. The idf term is a logarithmically scaled fraction that penalizes terms that occur frequently across the entire corpus. The rationale is that if a term t occurs in every document d of the entire corpus D, then that term is not informative because it is a commonly used term across all documents.

$$idf(t, D) = log \frac{N}{|\{d \in D : t \in d\}|}$$

$$tf(t,d, D) = tf(t,d) \times idf(t,D)$$

Once we have a set of features in real number vector space that represents each document, in theory we could run a K-means algorithm on each document based on tf-idf values for each term. However, the term-document matrix M is likely to be very sparse, since the dimensionality of the features is likely to be very high. Creating document clusters based on these features would encounter the 'curse of dimensionality' problem, because if most values in the feature vector for document d_n and d_m are 0, then the K-means algorithm may categorize them in the same cluster even though they are not, in fact, topically similar.

2.3.3 Latent Dirichlet Allocation

One of the most basic forms of topic modelling is Latent Dirichlet Allocation (LDA). LDA is a generative probabilistic model that treats each document in a corpus as a mixture of latent topics, or hidden variables, that are generating the tokens (terms) that constitute the bag of words. Topic modeling fulfills the content organization function of GSumm and is ideal for dealing with unstructured data. Our rationale behind using LDA is that it is also a well studied and well documented technique, and has several open source implementations in Python, which makes its implementation relatively straightforward. We used the Gensim package written by Řehůřek,

For each document d_i, an LDA model would return a vector of probabilities that represent the degree to which each topic k (where k = 1, 2, 3 … K topics) contributes to the topical distribution of d_i. LDA specifies a joint probability distribution over the observed variables, which is the frequency of terms in each document, and the hidden random variables. The model can then compute the conditional distribution of hidden variables given the observed variables. In the context of text analysis, these observed variables are tokens, or instances of words in a document. We can formulate this joint distribution as the following:

$$p\left(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}\right) = \prod_{i=1}^{K} p\left(\beta_i\right) \prod_{d=1}^{D} p\left(\theta_d\right) \left(\prod_{n=1}^{N} p(z_{d,n} | \theta_d) p(w_{d,n} | \beta_{1:K}, z_{d,n}) \right)$$

Where $\beta_{1:K}$ are k topic and each β_K is a distribution over the fixed vocabulary of words, the topic proportions for the dth document are Θ_d, where $\Theta_{d,k}$ is the degree to which topic k contributes to the topic distribution of document d. Each word in the dth document is also has a topic assignment, $z_{d,n}$ where n is the nth word in document d. All these random variables thus far are hidden. The observed variable is the observed word in the document d, notated as w_d, where $w_{d,n}$ is one of the words in the fixed vocabulary of types. For a visual representation of this joint probability in plate notation, see Appendix Figure 1.

2.4 Document Clustering

2.4.1 Divide into clusters

From a cluster of texts provided by the LDA topic modeling, which represents a certain topic, or set of topics, we would like to create a compressed version that provides as much information to the user as possible about that cluster.

In order to do this, we divide the cluster of texts further, into *n* sub-clusters, with *n* representing the number of sentences we wish to be returned. As this is another unsupervised clustering problem, we could use similar methods to topic modeling (i.e. K-means, Gaussian Mixture Models). We simply need a method that separates groups of sentences from each other, so that they are as unlike each other as possible. This will allow for the final returned sentences to touch on several different topics. For the current study, we used K-means and specified *n* to have a value of 5.

2.4.2 Judging sentence centrality

From there, it is a matter of extracting the sentences in each sub-cluster that are most central. There exists a few different methods to find the most central sentences.

We used *graph-based summarization*, which is similar to centroid-based summarization, with a few alterations. In *centroid-based summarization*, the sentences that contain more words from the centroid of the cluster are considered important. Inspired by social network analysis, graph-based summarization uses a graph representation of a document cluster, where vertices represent the sentences and edges are defined in terms of the similarity relation between pairs of sentences.

A cosine similarity matrix can be constructed to tabularly represent the network. The cells of the cosine similarity matrix, or the similarity between sentences, can be calculated with the following formula:

$$\text{idf-modified-cosine}(x, y) = \frac{\sum_{w \in x,y} \text{tf}_{w,x} \text{tf}_{w,y} (\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x} (\text{tf}_{x_i,x} \text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y} (\text{tf}_{y_i,y} \text{idf}_{y_i})^2}}$$

If we are interested in only the most significant similarities, we can define a threshold for similarities. Relationships between sentences with similarities that fall under this threshold will be equated to zero. After a cosine similarity matrix is constructed, the sum of all these values for each sentence can be calculated to represent the "prestige" of a sentence.

Using this framework, the centroids are the sentences that are of the highest prestige in the network of sentences (prestige implies centrality in a social network framework). In other words, the sentences that have the highest similarity to all other sentences are the centroids.

3. Results

3.1 Corpus Creation

We queried the PubMed Open Access API to scrape 1000 articles from the Biomed Central Public Health Journal. After preprocessing and cleaning, the corpus of 1000 documents had 19,613 features, with 674,260 non-zero entries.

3.2 LDA Topic Distributions

In accordance to common practice, we chose to create an LDA model where K = 20 latent topics based on our corpus. Refer to Figure 2 in the Appendix for per-word probability distributions.

3.2 Multi-document Sentence Extraction

We ran our topical LDA model on the article written by Endevelt, et al., called 'An intensive family intervention clinic for reducing childhood obesity', which we obtained by searching for 'childhood obesity' on the native PubMed search bar. Refer to Supplement 1 in the Appendix for the results of the automatic literature review.

4. Conclusion

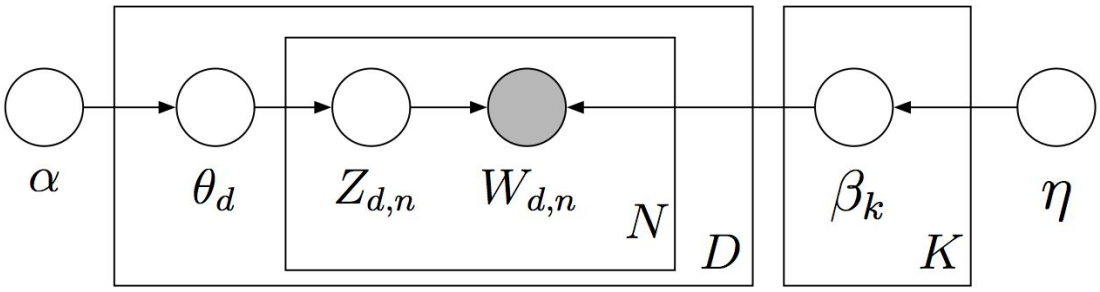
Although we did not create a complete product, we researched into the methods from which a product could be created. From this project, we successfully identified a means to organize articles using topic modeling. We also successfully extracted the most important sentences from a cluster of articles around a certain topic. Possible future directions include using MESH terms to attempt to improve the feature selection and preprocessing step, generating abstractive summaries from the extracted sentences to generate intelligent, human-readable sentences, and using an extension of hierarchical LDA to create a hierarchical structure of latent topics to further organize health science journal articles.

References

1. Bird S, Klein E, Loper E. *Natural Language Processing with Python*. Sebastopol, CA: O'Reilly Media, Inc.; 2009.
2. Blei DM, Ng AY, Jordan MI. Latent Dirichlet Allocation. *Journal of Machine Learning Research*. 2003;3:993-1022.
3. Blei DM, Lafferty JD. Topic Models. *Text Mining:: Classification, Clustering, and Applications*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2009.
4. Chowdary CR, Kumar PS. ESUM: An Efficient System for Query-Specific Multi-Document Summarization. *Advances in Information Retrieval - 31st European Conference on IR Research*. Toulouse, France, 2009.
5. Endevelt, R., Elkayam, O., Cohen, R., Peled, R., Tal-Pony, L., Grunwald, R. M., ... & Heymann, A. D. An intensive family intervention clinic for reducing childhood obesity. *The Journal of the American Board of Family Medicine*, *27*(3), 321-328, 2014.
6. Erkan G, Radev DR. LexRank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research*. 2004;22.
7. Mabe M, Ware M. *The STM Report: An Overview of Scientific and Scholarly Journal Publishing*. The Netherlands, 2012.
8. McKeown K, Radev DR. Generating summaries of multiple news articles. *Proceedings, 18th Annual International {ACM} {SIGIR} Conference on Research and Development in Information Retrieval*. Seattle, Washington1995:74-82.
9. Murphy KP. *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: The MIT Press; 2012.
10. National Institutes of Health. Medical Subject Headings. 2013.
11. Radev DR, Jing H, Stys M, Tam D. Centroid-based summarization of multiple documents. *Information Processing and Management*. 2004;40:919-938.
12. Řehůřek R. Gensim Python Package: Topic Modelling for Humans. 2014; http://radimrehurek.com/gensim/.

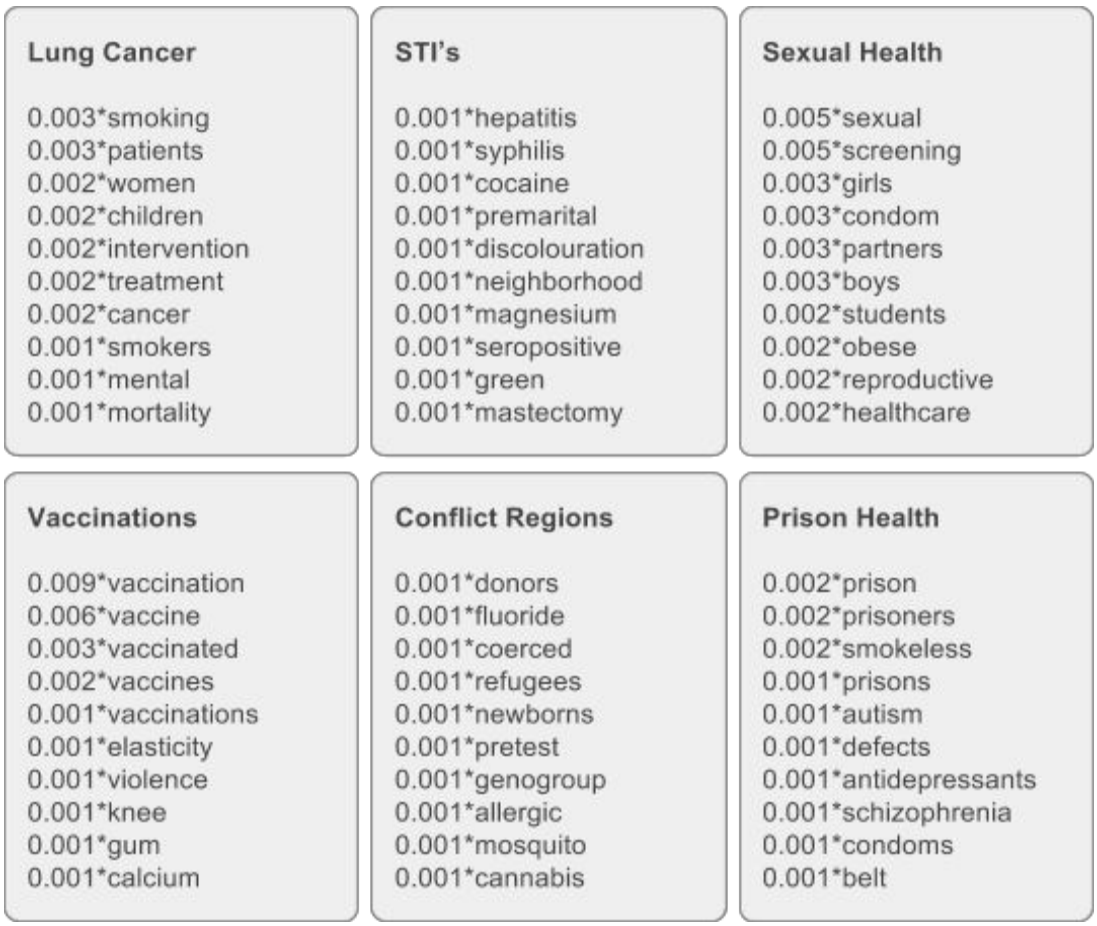
Appendix

Figure 1: LDA Joint Probability in Plate Notation



Nodes represent random variables. Shaded nodes are observed, non-shaded nodes are hidden variables, and edges represent dependencies between random variables. Taken from Blei & Lafferty 2009.

Figure 2: LDA Topic Distribution



Six selected topics from the K = 20 that were interpretable to a human-readable 'topic'. Shown below the hand-labelled topic are the words with the top 10 probability values in the fixed vocabulary based on the unique tokens of the corpus

Supplement 1: Multi-document Summary Results

Results of our LDA clustering and sentence extraction pipeline. We clustered our corpus of 1000 articles around the article by Endevelt et. al (2014) and then extracting the most 'central' sentences within the topical cluster.

Sentence 1:
Table 2 below presents summary information showing the difference in risk factor levels between the two ethnic groups, for the two most distant time periods in the study (19891990 and 19979), along with the proportion ascertained and the number of individuals contributing data to each time period and for each individual risk factor.

Sentence 2:
Being overweight or obese can have a significant impact on health, as these individuals are more likely to suffer from a variety of illnesses [1,2,10,11,13], have an increased risk of mortality [14-16], and use more health care resources and disability days [11] than their normal weight peers.

Sentence 3:
In regression analysis, adjusting for age and test year, South Asian men and women throughout the study period appear to have comparatively either lower or similar risk factor levels, with the exception of BMI for women (Additional File 4, Figures 1-10).

Sentence 4:
Five of six studies reported associations between low early-life SES and little or no adult leisure-time physical activity [14,61,63,64,66], five of five found associations with high adult alcohol intake [52,61,63,68,73], and eight of twelve studies reported associations with higher smoking rates as study findings [14,44,52,63,65-68].

Sentence 5:
However, we must be careful in those interpretations, because this study has the disadvantage of being a self-reported data study, where the true prevalence of overweight and obesity is expected to be underestimated by men and women in almost all ages [37,38].