Automatic Evaluation of Topic Model Labels using Wikipedia Categories

by

Jonathan Snyder

B.S., University of Colorado at Colorado Springs, 2007

A thesis submitted to the Graduate Faculty of the

University of Colorado at Colorado Springs

in partial fulfillment of the

requirements for the degree of

Master of Science

Department of Computer Science

2016

This thesis for the Master of Science degree by

Jonathan Snyder

has been approved for the

Department of Computer Science

by

Dr. Jugal Kalita, Chair	Date
Dr. Rory Lewis	Date
Dr. Terrance Boult	Date

Abstract

Manual summarization is extremely labor intensive. Recent research has shown that topic models can provide a good overview of unstructured data. Admittedly, the output of topic models can be hard to digest without good labels. Previous approaches for automatic topic model have used many different methods for creating candidate labels. For example, researchers have used Wikipedia article titles, keywords from article abstracts, or Del.icio.us tags as candidate labels. In every case the labels were reviewed using human evaluators. The contribution of this thesis is of an automatic method for evaluating topic labels. Different labeling methods are evaluated using this automatic evaluation metric. It is found that short noun phrases extracted from the original text give valuable context to topic models.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Classifying responses can be a useful text mining technique. Traditional classification approaches are labor-intensive [1], but topic models [2] have recently been shown to be a powerful unsupervised tool to identify latent text patterns.

Topic models seek to capture the latent topics inherent in the data. Using information related to the co-occurrence of words, topic models tease out which words occur frequently with which other words, and group them into topics. The result of this computation is that each word in the corpus is assigned to a topic. Analysis is done to determine the top occurring words for each topic, and the topic distribution for each document. From this, a user can understand what the top occurring ideas are in the documents, and also browse the documents containing those topics. One of the problems with this, however, is that the title for a topic model is simply a list of words. This is not a friendly label for a user to digest, and requires an understanding of topic model theory. A better way to accomplish this would be to provide a title for the topic. This thesis explores a new method for the automatic

Frequency	Word
2193	galaxy
1852	love
1711	samsung
902	note
841	2
607	lii
552	mobile
351	awesome

Table 1.1: A Topic Extracted from Samsung Mobile's Facebook Page

labeling of topic models.

Topic models can provide a nonhierarchical soft clustering of the documents using the document-topic probabilities. Labeling the document according to the most represented topic results in a hard clustering. In the domain of Facebook brand pages, topic models can be used to understand the concerns and general feelings of the page's fans. The words that occur most frequently in a topic provide a representation for the ideas that are expressed in the topic. For example, Table 1.1 and Table 1.2 lists some of the most frequently occurring words in two topics extracted from Samsung Mobile's Facebook page. The topic in Table 1.1 most likely represents customers happy with Samsung phones, and the topic in Table 1.2 most likely

Frequency	Word
1159	phone
493	battery
467	time
368	work
304	sprint
230	back
210	problems
187	apps

Table 1.2: Another Topic Extracted from Samsung Mobile's Facebook Page

represents customers complaining about battery life issues, but it is hard to know if that labeling is correct with the limited information of just the top topic words.

1.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [3] is a standard algorithm for topic modeling. LDA uses a probabilistic generative model, which requires very little human intervention. The idea in LDA is that each document is composed of a set of topics. The probability of a topic occurring in a document follows a Dirichlet distribution which favors giving a few topics high probability with the others receiving low probability. Each topic then has a probability distribution of words. See chapter **??** for a more detailed description of the theory behind LDA. Continuing the example from Facebook data, the word 'battery' has a high probability of occurring for the second topic example given previously. The following gives an example of a post. The numbered superscripts on some of the words indicate that they belong to one of the previous two topics:

> I love[1] Samsung[1] with a passion...i had the galaxy[1] II[1] but the battery[2] died really quick...it stopped charging[2] completely[2] so I had to send it away.

LDA has extracted 2 topics from the sentence. Topic 1 having to do with loving the phone, and Topic 2 having to do with battery life. The other words in the sentence were filtered out as stop words or part of other topics.

1.2 General Approach

One of the difficulties with research on automatic labeling is that there is how to measure the accuracy of a generated label. Most research uses human annotators to measure the quality of the generated labels. Indeed, looking at the top words for a topic may not be enough to capture the nuance of the topic. It may be necessary to look at the documents containing those topics.

In addition to the large corpus of articles Wikipedia provides, it also contains a hierarchy of categories. The categories can be used to simulate a topic model, and can be matched to a topic model found in the same dataset. In this thesis, a Wikipedia dataset is used to generate a set of gold standard category labels against a set of Topic Models generated from the same dataset.

1.3 Contribution

The contribution to the computer science body of knowledge is an automated method for measuring the quality of generated topic model labels. Methods detailed in other research are applied to the dataset to test the validity of the method.

1.4 Overview

Chapter 2 details the related work in opinion mining, topic modeling, and the automatic labeling of topic models. Chapter 3 gives theoretical background of Latent Dirichlet Allocation including mixture models, generative models, Dirichlet distri-

butions. The implemented labeling methods are outlined in chapter 4. Chapter 5 describes in detail the automated method for measuring the quality of the generated topic labels. Finally, chapter 6 details the results of the labeling methods.

Chapter 2

Related Work

2.1 Opinion Mining

Opinion Mining is the process of collecting and categorizing opinions about a product. This area of research is closely related to topic models. Jackson et. al. [1] present methods for manually tagging and clustering user responses. Their methods involve having 10 people manually cluster survey responses, then combine their clusters using multidimensional scaling [4] and k-means clustering [5]. Finally the researchers decide on the appropriate number of clusters and approve cluster labels. This method creates good rigor and eliminates some bias by not creating a tagging scheme before hand, but is quite labor intensive.

Product reviews can be an especially good source of information, and supply a large dataset for research. Lee et al. [6] analyzed reviews on specific products to extract product features and then detect the sentiment of those features. Gamon et al. [7] describes a system to extract keywords from product reviews and then

detect the sentiment around those keywords. The keywords are then shown in a treemap and color coded according to sentiment. Zeigler et al. [8] built a similar system, but applied it to online survey responses. Although detecting sentiment can provide important information from customer feedback data, a first step is to find good ways of extracting the product features or concerns people are having. The opinion mining methods here focus more on sentiment analysis and suffer from problems related to synonymy and polysemy. These papers used term frequency-inverse document frequency [9] metrics to extract key words and phrases from the text.

Ganisen et al. [10] developed a tool called Opinosis which seeks to create abstractive summaries from sets of sentences. The domain they tested on was product reviews. They created a graph representation of all the sentences for a particular topic and then found the most used path in the graph that parses as a complete sentence.

2.2 Topic Modeling

Simply understanding the topics can be a large part of understanding a users concerns in a customer feedback situation. Indeed, Rilof et al. [11] find that topic-filtering and subjectivity filtering are complimentary (i.e. that topics generally contain a cohesive sentiment). In a study of various opinion mining systems, Pang [12] found that Topic extraction is important for documents containing (1) comparative studies or (2) discussions of various features, aspects, or attributes.

Lee et al. [13] studied 4 text mining methods including Latent Semantic Analysis (LSA) which analyzes which words occur frequently together to find key phrases and topics; Probabilistic Latent Semantic Analysis (pLSA) which introduces probability theory to determine one topic for every document; Latent Dirichlet Analysis (LDA) which extends the probabilistic model to allow for one document to contain a set of topics with a probability distribution; and Contextual Topic Modeling (CTM) [14] which extends LDA to use a logistic normal distributions instead of dirichlet.

Hong et al. [15] applied LDA to Twitter messages to see if it could predict the topics twitter users would write about in the future. LDA proved to be useful compared to other methods. A large part of the research was finding the best way to model documents. For example, should the document be defined as each Twitter message or an aggregate of all of a users messages? Hong et al. found that larger documents provided better results.

Recent work in the field of LDA research has focused on extending the LDA to model more than just topics, documents, and words. Mei et al. [16] added sentiment to the model so that each topic contains positive, negative, and neutral distributions of words. Xia [17] extended the LDA model by adding user information. Chang et al. [18] added the idea of a sentence to the LDA model.

2.3 Labeling Topic Models

Mei et al. [19] was one of the first to tackle the problem of labeling topic models. He extracted a set of candidate labels from a representative corpus. Then he created a distribution of words for each candidate label using the words that occurred around the candidate labels in the representative corpus. Using these distributions of words, he matched the labels to the topic models. Mei evaluated a label using manual surveys of label preferences. Mei was able to achieve good results.

Indeed Mei's methods have been applied to other domains with success. Magatti [20] used Google Directory names as candidate labels with the content of the pages as the representative distribution of words. Lau et al [21] used Wikipedia article titles as candidate labels with the articles content as the representative distribution of words. Xia [22] used scientific abstracts as the representative distribution of words, and the abstract keywords as candidate labels. Ramage [23] used Del.icio.us web page tags.

Chang et al. [18] used topic models to find the most representative sentences for a particular topic model, and use that to provide an extractive summary of the documents. Although Chang et al. wasnt specifically trying to create labels for topic models, their approach could be helpful in finding good labels for topic models.

Chapter 3

Latent Dirichlet Allocation

Latent Dirichlet allocation uses a generative mixture model to find latent topics in the data. This chapter goes into detail of how this model works.

3.1 Generative Models

A generative model places probabilities on both the observed data, and the hidden parameters of the model. Generative models can be contrasted with discriminate models. Discriminate models instead seek to directly determine which class a particular observation belongs to. Discriminate models include logistic regression models and support vector machine. These models try to learn how to discriminate between classes. On the other hand, generative models create a story for how the data was generated and then try to find the probabilities that lead to the highest chance of seeing the observed data. In the example given in the last section, the story was that the documents were created using this story:

1. Decide on an author A_i that will write the document using $P(A_1)$ and $P(A_2)$
2. Generate 3 words in the document using the author's word probabilities, $P(V \mid A_i)$
3. Repeat for 4 documents

3.2 Mixture Models

In a mixture model, there are a set of observations with the assumption that these observations come from a set of probability distributions. The task is to determine the most likely probability distributions and also the assignment of each observation as coming from one of the probability distributions.

As an example in the realm of natural language processing, suppose there is a set of 100 documents known to be from two different authors; however, the mapping between authors and documents is missing. The task of the mixture model is to re-label the documents into the two authors. First, we make the assumption that the labels can be determined simply by using a bag of words model (i.e. the word order and sentence structure does not matter.) Next we randomly generate two probability distributions over all the words in the vocabulary. That is we assign a probability that an author will use a particular word in the vocabulary.

3.2.1 Assign Authors

Next we look at each document and compare the words used in that document against the probability distribution for each author. At this stage we assign a weight between 0 and 1 of the likelihood that the author wrote that document. Indeed this the intended output of the algorithm; however, the data most likely is not useful as the author models were generated randomly.

3.2.2 Adjust Author Model

The next stage is to adjust the author model. Using the weight generated in the last step, a weighted average is created for each author. This step is the solution to the following problem: given these documents and weights, what is most likely to be the author's probability distribution over words? Assigning authors and the adjusting the author model is repeated until the probabilities converge.

3.2.3 Example

This section details a small toy-sized example of a mixture model containing 4 small sentences with two authors. D_i is a document. Each document contains 3 words.

$$\begin{aligned} D_1 = [i, like, cats], D_2 = [cats, are, cool], \\ D_3 = [i, like, dogs], D_4 = [dogs, are, cool] \end{aligned} \tag{3.1}$$

The vocabulary consists of the 6 words used in the documents.

$$V = [i, like, cats, are, cool, dogs] \tag{3.2}$$

There are two authors, A_1, and A_2. Each author has a probability distribution over the words in the vocabulary. The probability distribution is initialized randomly.

$$\begin{aligned} P(\, V \mid A_1 \,) &= [0.16, 0.20, 0.07, 0.20, 0.18, 0.18] \\ P(\, V \mid A_2 \,) &= [0.13, 0.21, 0.13, 0.26, 0.03, 0.24] \end{aligned} \tag{3.3}$$

Additionally, the prior probability for each author is set to 0.5. This corresponds to the assumption each author wrote half of the documents.

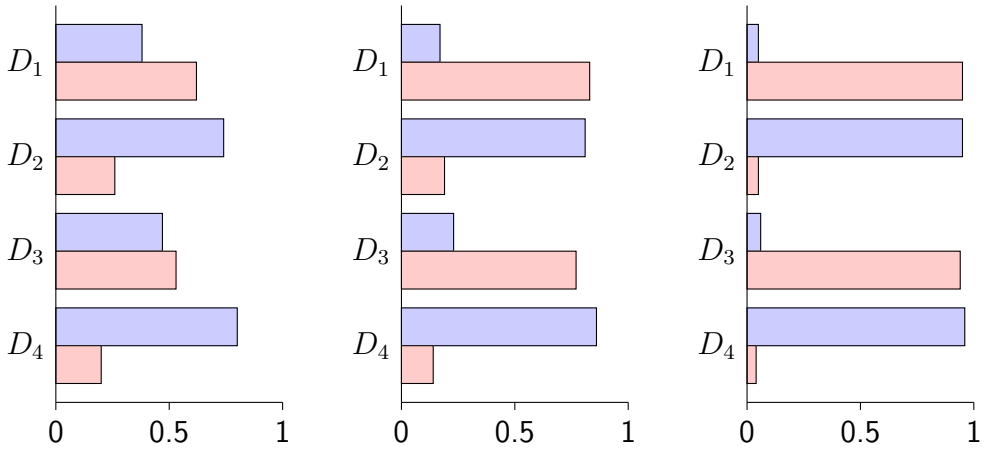
$$P(A_1) = P(A_2) = 0.5 \tag{3.4}$$

The probability that each document is produced by each author is determined:

$$P(\, D_i \mid A_j) = \prod_k P(\, D_{ik} \mid A_j \,) \tag{3.5}$$

Next, using Bayes' law, the probability of an author given a document is determined with:

$$P(\, A_j \mid D_i \,) = \frac{P(\, D_i \mid A_j \,)\; P(A_j)}{\sum_k P(\, D_i \mid A_k \,)\; P(A_k)} \tag{3.6}$$



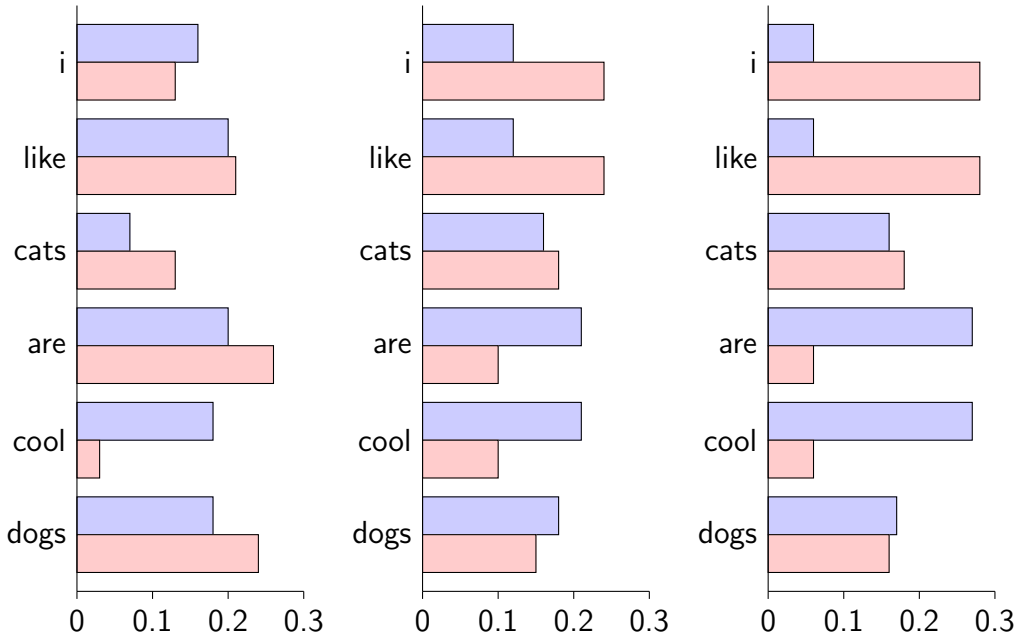
(a) Iteration 1 (b) Iteration 2 (c) Iteration 3

Figure 3.1: $P(\, A_j \mid D_i \,)$ across 3 iterations (A_1 is the top bar of each set)

Then, each author's probability distribution over words is updated. This is basically a weighted average using $P(\, A_i \mid D \,)$ as weights.

$$P(\, V_i \mid A_j \,) = \frac{\sum_k P(\, A_j \mid D_k \,)\; P(\, V_i \mid D_k \,)}{\sum_k P(\, A_j \mid D_k \,)} \tag{3.7}$$

Finally, equations 3.5, 3.6, and 3.7 are repeated until a suitably useful result is achieved. Figures 3.1 and 3.2 show the progress of this algorithm over 3 iterations. Looking at 3.1c, A_1 appears to have written D_2 and D_4. This matches with the writing style of the sentences. Although "I like cats.", and "Cats are cool." express similar ideas, "Cats are cool.", and "Dogs are cool." posses a similar writing style. Another interesting feature to the result is that the probability of "dogs" and "cats" occurring is independent from the author (i.e. both authors use those words with equal probability)



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3

Figure 3.2: $P(V_i \mid A_j)$ for 3 iterations (A_1 is the top bar of each set)

3.3 Dirichlet Process

To understand a Dirichlet process, first consider a Bernoulli process which is a sequence of independent random events that take the form of two discrete values. It can be understood as the result of repeatedly flipping a coin and recording the results. A Bernoulli distribution is the probability distribution of this process. For example, if after flipping a coin 10 times and 7 are heads, the Bernoulli distribution can be used to determine the probability of this occurring.

Just as a Bernoulli distribution can be understood as the result of coin flips, the Dirichlet distribution can be understood as a random process. Consider an urn containing balls of K different colors. Initially the urn contains α_1 balls of color 1, α_2 balls of color 2 etc. Now draw a ball from the urn at random. Return the ball to the urn, but also add an additional ball of the same color. If this process were

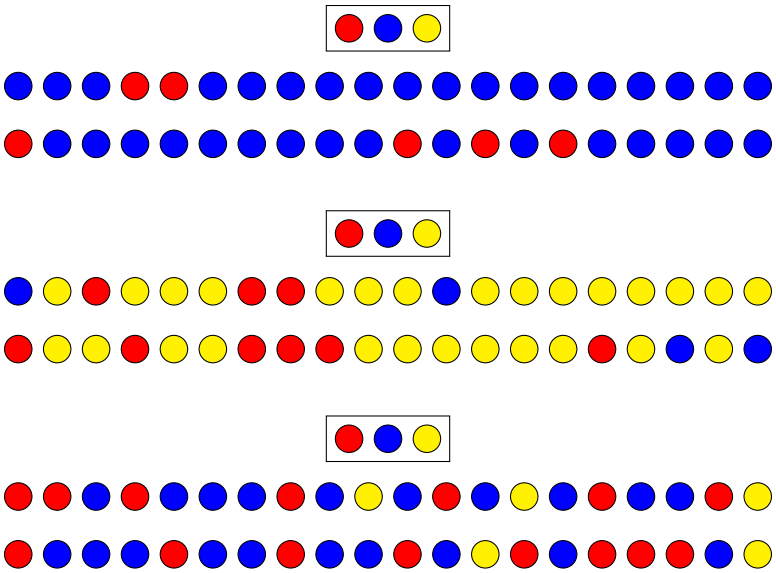


Figure 3.3: Dirichlet Process with $\alpha = (1, 1, 1)$

repeated forever, the distribution of balls in the urn would approach a Dirichlet distribution with parameters $\alpha_1, \alpha_2, \ldots, \alpha_K$.

Figures 3.3 and 3.4 show examples of a dirichlet process carried out where $k = 3$. Figure 3.3 shows 3 examples with $\alpha = (1, 1, 1)$, and figure 3.4 show 3 examples with $alpha = (6, 6, 6)$. The box in the figures indicates the starting configuration of the urn, and the other circles represent the balls that were added to the urn. These examples help to give some intuitive understanding of the kinds of distributions that arise. If the urn starts with just one ball of each color, one color will many times dominate the distribution. Indeed the first example in figure 3.3 shows a yellow ball was never chosen because blue is dominating the distribution. Conversely, if the α values are increased, the distribution is more even as shown in figure 3.4.

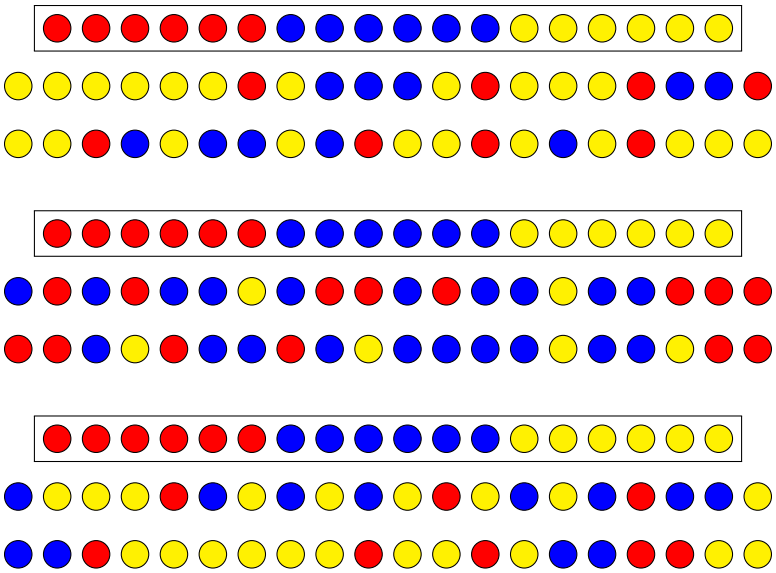


Figure 3.4: Dirichlet Process with $\alpha = (6, 6, 6)$

3.4 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) combines these concepts to find the latent topics in the documents. It accomplishes this with a probabilistic generative model using Dirichlet probability distributions. The generative process involves these steps when creating a document:

1. Decide on a distribution of topics for the document
2. Select a topic from the distribution of topics for the document
3. Select a word from the distribution of words in the topic
4. Repeat this process until the document is complete

The algorithm iterates in much the same way as described in the example mixture model above, except now the topic models add a layer of complexity. Additionally, all the probability distributions are modeled after a Dirichlet process. Us-

ing a Dirichlet process leads to documents containing only a few high probability topics. Similarly, the topic distributions are skewed to contain a small amount of high probability words.

Chapter 4

Labeling Methods

This chapter discusses the methods used to prepare a set of documents, generate topic models, and the creation of topic labels.

4.1 Documents Creation

Wikipedia provides a huge corpus of human annotated text. DBpedia [24] is a knowledge based extracted from Wikipedia. Using the data dumps from DBpedia made the process of parsing the Wikipedia data much easier. There is a data dump called "Long Abstracts" that holds the first, introductory paragraph of an article. See appendix A for examples of the input from DBpedia. Each long abstract is used as a separate document.

In addition to the long abstracts, DBpedia also provide data dumps for just about every part of Wikipedia. Of interest here is the category hierarchy. The data dumps for the category hierarchy first list all the categories, giving the full name of

the category. Second the data dump lists the hierarchical relationship among the categories and the articles that are in each category.

One complication to this data is that content creators on Wikipedia will also combine or rename articles and leave a redirect in its place. When creating the category and article hierarchy, many times the exact article is not present in the list of abstracts, but after following a redirect the referenced article can be found.

Using all the articles in wikipedia would result in a very large dataset of millions of documents. A subset of the articles is prepared by starting with a hand selected category, and following the category hierarchy down a configurable number of steps, including all the articles referenced by those categories. One other complication with this data, is that the categories may contain cycles. Care is taken to only include an article once in the subset of data.

4.2 Text Preparation

Sentence splitting and tokenization is preformed using the NLTK: the natural language toolkit [25]. Sentence splitting is preformed so that when extracting candidate labels from the text, a candidate label is not chosen across sentence boundaries. Conversely, the LDA topic model calculation just uses a bag of words model so the sentence boundaries are unnecessary for its calculation.

Tokens are prepared by first transliterating non-ASCII characters into their ASCII equivalents. For example changing vowels with diacritics (i.e. resumé becomes resume.) In testing it was discovered that the diacritics were applied in-

consistently, resulting in topics containing all versions. Next punctuation is discarded. This is mainly to correct inconsistently hyphenated words. (i.e. "long-term" becomes "long term"). Finally, tokens containing non-alphanumeric characters are discarded. This removes other strange tokens such as urls. Most of these discarded tokens only appear once, and because LDA is mostly concerned with the co-occurrence of words, discarding these tokens does not change the effectiveness of the LDA calculation. The original text is kept to prevent candidate labels from spanning across discarded tokens, and to be able to revert candidate labels back to their original form.

Stop words are then removed. Although LDA does not require the removal of stop words, sometimes it would generate an entire topic made of mostly stop words. For example a topic was found once including the words "first", "second", "third", etc. While this topic is interesting from the standpoint of analyzing the algorithm, it is not helpful in getting a general summary of the documents.

4.3 Latent Dirichlet Analysis

The topic models are generated using MALLET [26], a statistical natural language processing package written in Java. MALLET's LDA tool takes a number of parameters including the number of topic models to generate. The output of the processing is a mapping of every input word into it's topic. This output is parsed and saved to the model. A discussion of the hyper-parameters used is including in Section 4.8.

4.4 Candidate Labels

As mentioned in the introduction, the creation of candidate labels from the text of the documents is one of the goals of this Thesis. The following subsections describe methods for generating candidate titles:

4.4.1 Chunked Noun Phrase

This is the method Mei et al. [19] used as mentioned in section 2.3. First, a list of candidate labels are generated by extracting the chunked noun phrases from the document. The chunking uses NLTK's part of speech tagging functionality.

4.4.2 N-Gram

A brute-force method of candidate label generation is to simply use all N-grams of words from the input text. This can generate a very large list of candidate labels. Depending on the method used to evaluate the candidate labels, this could be prohibitively expensive. If the evaluation metric is easy to calculate; however, this can ensure that no good labels are missed when using more restrictive methods such as noun phrase chunking.

4.5 Ranking Candidate Labels

The following subsections describe many methods for choosing and ranking candidate labels against the topic models.

4.5.1 KL Divergence of Co-occurring Words

This methods collects all the words appearing close to the candidate label in the text. A label is ranked according to how closely the distribution of the co-occurring words matches a topic model. Mei et al. [19] used the Kullback-Leibler divergence to determine which distribution matches.

4.5.2 Term Frequency

Looking at the top 10 words for a topic, it is tempting to just combine those words into a meaningful phrase, and use that as the model. One way to do this is to score the labels just looking at the words present in the label. Simply count the number of times the words in the candidate label are mapped to a topic.

4.5.3 TF-IDF

A good candidate label will not only describe the topic model, but it should also be distinct from other topics. One way to do this is to borrow the concept of inverse document frequency (IDF) from information retrieval research and compute the inverse topic frequency. This evaluation metric, looks at each word individually, and computes the number of times it occurs in the topic, then divides that by the log of the percentage of topics that contain that word. The score of the label is then the sum of these scores.

4.5.4 Cosine Similarity

One problem with summing term frequencies, is that it can favor long labels. Cosine similarity is a method of normalizing document length. This method is to compare the words in the label with the frequency of words in the topic.

4.6 Cosine Similarity Performance Optimization

Giving a score to every candidate label for every topic can be time consuming. One way to reduce the time required to find the best labels is to use a branch and bound algorithm. First all the candidate labels are organized into a prefix trie where each node in the tree is a word. A brute force method would be to visit every node in the trie calculating the score of that node. Using the branch and bound algorithm can significantly reduce the number of nodes that need to be visited. For example, if a cosine similarity is used the score of topic t and label l, $S_{t,l}$ is:

$$S_{t,l} = \frac{\sum_{i=0}^{n} tf(w_i, t) tf(w_i, l)}{\sqrt{\sum_{i=0}^{n} tf(w_i, t)^2} \sqrt{\sum_{i=0}^{n} tf(w_i, l)^2}} \tag{4.1}$$

where $tf(w_i, t)$ is the term-frequency of word w_i in topic t, and $tf(w_i, l)$ is the term-frequency of word w_i in label l. The term in the denominator for the length of the the topic vector can be ignored because it will be the same for all candidate labels, and therefore does not affect the ranking. To bound the number of nodes visited the maximum score of a candidate label prefix is calculated. Using the length of the longest candidate label N, and the largest term frequency for the

topic: $max_i tf(w_i, t)$, the assumption is made that terms will not be repeated, each additional term will have the largest term frequency, and the candidate label will be of length N.

$$\max S_{t,l_p} = \frac{\sum_{i=0}^{n} tf(w_i, t) tf(w_i, l_p) + \max_i tf(w_i, t) * (N - p)}{\sqrt{N}} \tag{4.2}$$

If the trie is visited in descending order according to the topic word frequencies, when a bound condition is reached the rest of the children of the current node can also be skipped.

4.7 First Order Preprocessing

During initial testing, the method that Mei use in [19] produced poor labels when using a phrase's sentences as its context. The exact methods Mei used were not entirely clear from his paper. This method was modified to first reduce the number of labels under consideration by choosing the top ten labels that match the topics according to the cosine similarity of the word-topic frequency. Then Mei's methods were used to choose the best label from that set.

4.8 Discussion of Hyper-parameters

Subset Levels The number of levels deep to select articles and categories from. This was set using trial an error until a suitably sized dataset was achieved for each of the starting categories.

Dataset Size The size of the dataset was increased until the dataset could no longer run comfortably in memory.

Starting Categories "Computer Science", "Culture", and "Popular Music"

Number of Topics 100, and 30 - Most of the papers used 30 topics. Using higher and lower numbers of topics helps to compare the results against other topic sizes.

Number of Iterations 2000 - After about 2000 iterations, there is little benefit to running more.

Use Symmetric Alpha false - Setting this to false allows there to be un-evenly sized topics. This generally provides more coherent topics as one would not expect all the actual topics to be of the same size.

4.9 Corpus Statistics

Table 4.1 lists some statistics from the Wikipedia dataset that was used to run these experiments.

	"Computer Science"	"Culture"	"Popular Music"
Subset Levels	5	2	4
Articles	51,799	37,860	78,204
Categories	1,427	750	2,361
Words	6,661,293	5,586,848	8,254,562

Table 4.1: Corpus Statistics

Chapter 5

Evaluation Methods

As mentioned in the introduction, evaluation can be a tricky part of labeling topic models. What makes a good label, and how can it be measured? A large part of this thesis was finding a good way to measure labeling performance automatically. This is where the Wikipedia dataset is great. It has a large set of categories that can be used as a surragate topic.

5.1 Matching Articles with Topics

The first problem is how to assign a set of articles to each topic. After the topic modeling runs, the output is an assigned topic to each word. One way is to choose the most representative topic for an article. This has the problem of excluding many articles from topics. A fix for this could be to choose the top N most representative topics for an article. Consider an article where the topics are fairly represented among all the topics. Ideally, in this case, this article would not be assigned

to any of the topics. On the other side of the spectrum is an article containing words primarily from only 2 topics. Ideally, this article would only be assigned to those 2 topics even if N was greater than 2.

One method to overcome these difficulties is to use a test for statistical significance. Using the null hypothesis that the topics were chosen with equal probability, find the topics that are over-represented in the article. This would require finding the number of words in the article at which a topic becomes significant, and then assign the article to the topics which have that many words in the article. This is done using a binomial probability test, which is to find the smallest n such that

$$0.95 < \sum_{k=0}^{n} \binom{N}{k} p^k (1-p)^{N-k} \tag{5.1}$$

where N is the number of words in the article, and p is 1/(number of topics).

5.2 Matching Articles with Categories

The next problem is how to assign a set of articles to a category. This is computed by starting with a category and first finding all the articles directly beneath. Then finding all the categories directly beneath, and finding the articles of those categories. All the articles in the category are recursively discovered in this manner.

For example, figure 5.1 shows a small selection of categories and articles. There are 3 categories in this example, “Information Retrieval”, “Searching”, and “Vector Space Model”. There are 7 articles along the bottom of the figure, all in boxes. The

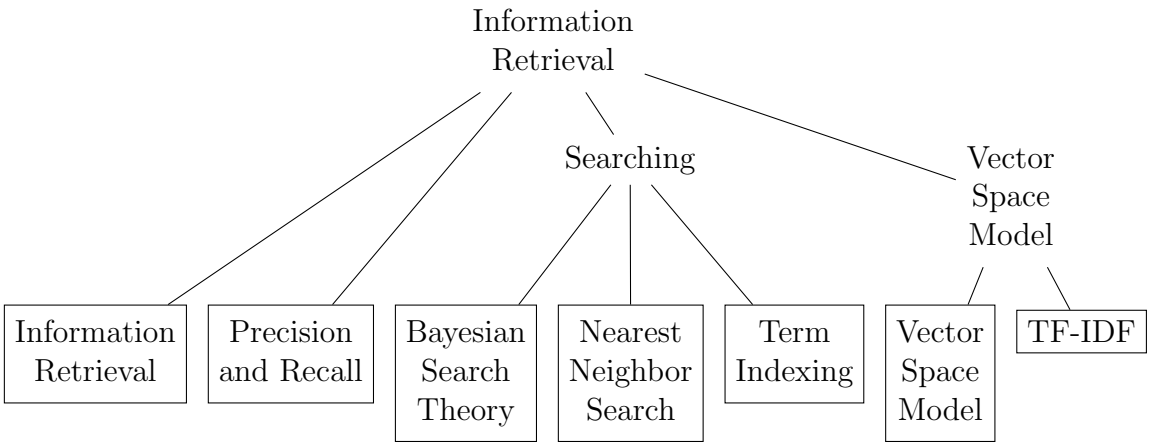


Figure 5.1: Example Categories and Articles

Article \ Category	Information Retrieval	Searching	Vector Space Model
Information Retrieval	1	0	0
Precision and Recall	1	0	0
Bayesian Search Theory	1	1	0
Nearest Neighbor Search	1	1	0
Term Indexing	1	1	0
Vector Space Model	1	0	1
TF-IDF	1	0	1

Table 5.1: Category Article Mappings from Figure 5.1

lines show the relationships among the categories and articles.

Table 5.1 shows the resultant mappings between the categories and the articles. The articles are listed on the left, and the categories across the top. A one indicating that the article is mapped to the category, and a zero indicating it is not.

5.3 Matching Categories with Topics

The previous section describes mapping a set of articles to each topic. Now the goal is to find a category that most closely matches a topic. Another statistical test can be used to determine if a category even matches a topic better than chance. First,

the expected number of matching articles E for the topic is calculated.

$$E = N_t \frac{N_c}{N_a} \tag{5.2}$$

where N_t is the number of articles in the Topic, N_c is the number of articles in the category, and N_a is the number of articles in the corpus. One way to think about this is that N_c/N_a is the probability that an article chosen at random is part of that category. Then that probability is multiplied by N_t to get E.

Next, the observed number of articles O that are in both the topic and the category is calculated. Topic category pairs are considered when $O > E$ and the following Pearson Chi-Squared test is true:

$$\frac{(E-O)^2}{O} > 6.64 \tag{5.3}$$

Where 6.64 is the critical chi-squared statistic for a 99% confidence interval with one degree of freedom.

5.4 Ranking Candidate Categories

Information retrieval methods can be used to rank the candidate categories. A topic is considered as a set of matching articles against the category. Precision, recall, and f-measures are then calculated from these categories. The β parameter of the f-measure can be used to bias toward more broad or more narrow category titles.

5.5 Evaluation of Labels

ROUGE-1 and ROUGE-2 metrics measured against the best matching category titles are used to evaluate the generated labels.

Chapter 6

Results

In general the methods used in the experiments do provide good labels. Following are some examples of the best and worst labels from the results. Although the automatic labeling methods discussed here do not provide perfect labels, 2 or 3 word phrases do provide useful information about the topic models.

6.1 Best Case Analysis

As part of the experiment setup, the gold standard labels were searched for in the text. This was to answer the question of how good could a labelers possibly be by using substrings from the document collection. In most cases, the best case analysis showed that over 80% of the gold standard labels were to be found in the document collection. This was a surprising result because many of the papers [CITATION] made the assumption that good labels would need to be found outside of the original document collection.

ROUGE-1	0.857
ROUGE-2	0.750
Topic	image, color, images, graphics, display, pixel, colors, screen, pixels, blue
Categories	computer graphics algorithms, image processing, computer graphics
Labels	computer graphics, image processing, digital image
ROUGE-1	0.000
ROUGE-2	0.000
Topic	robot, robots, robotics, robotic, control, human, motion, autonomous, mechanical, movement
Categories	robotics, embedded systems, classes of computers
Labels	mobile devices, autonomous robots, industrial robots

Table 6.1: Labels from the 100 topic Computer Science dataset.

ROUGE-1	0.833
ROUGE-2	0.667
Topic	culture, cultural, cultures, people, society, western, indigenous, identity, world, groups
Categories	cultural studies, cultural education
Labels	popular culture, cultural heritage, cultural studies
ROUGE-1	0.000
ROUGE-2	0.000
Topic	character, comics, fictional, comic, marvel, created, appeared, dc, man, universe
Categories	fictional scientists, science and culture
Labels	marvel comics, dc comics, comic books

Table 6.2: Labels from the 100 topic Culture dataset.

ROUGE-1	0.875
ROUGE-2	0.800
Topic	country, single, music, album, song, released, singles, american, chart, billboard
Categories	country music, american country music, country music songs
Labels	american country, american country music, country music
ROUGE-1	0.000
ROUGE-2	0.000
Topic	music, indian, film, singer, india, playback, films, tamil, songs, director
Categories	filmi, indian film singers, indian film score composers
Labels	popular music, classical music, world music

Table 6.3: Labels from the 100 topic Popular Music dataset.

On the other hand, there were some cases where it is understandable that the label isn't part of the text. For example, one of the gold standard labels was "video games by graphical style". It is not expected that this would be part of the text of any of the articles, but clearly indicates a category title. Indeed this would not make a good label for a topic model because the phrase "by graphical style" has no meaning with an un-ordered collection of documents. It may be worthwhile to remove from consideration categories which include prepositional phrases beginning with "by".

Not all labels that were not found in the text were inferior. Another example from the Computer Science dataset highlights a situation where a category label is good, but the substring is not found in the text. The specific label , "mathematics and computing colleges in england", is a good label, but could only be generated from the text collection through abstractive summarization methods.

6.2 Dataset Selection

Overall, the labelers performed best on the Computer Science data set. Next, the Popular Music dataset performed reasonably well, but the performance on the Culture dataset was sub-par. The problem with the Culture dataset highlight the importance of choosing a good dataset. The biggest issue with the dataset was its small size, particularly with the small number of categories that were chosen from for the gold standard. This was a consequence of the broadness of the category. Indeed, grabbing articles more than 2 levels deep gave too many articles for my ex-

perimental setup to handle. Another issue with the Culture dataset was that many of the gold standard labels were one word; however, the simple cosine similarity metric biased toward 2 or 3 word phrases.

Another problem with the Culture dataset highlights a problem with using the word-topic frequency to judge a label. For the topic "music,film,theatre,dance,performance,films,st the gold standard identified the label "entertainment", however that word does not appear in the top ten for the topic.

In some cases the labeler found a better label than what the gold standard claimed the best was. For the topic model, "information,media,online,social,software,web,management,digi the best gold standard label was "collaboration". Clearly the label that the labeler chose, "social media information", gives more semantic meaning, but the limitations of the depth of the Culture dataset keep there from being a "social media" category to choose from.

6.3 Frequency versus TF-IDF

No significant difference was found between using the word-topic frequency vectors versus the TF-IDF vectors. This may be the case because of the way that the topic models work. Because the word distributions in the topic models follow a Dirichlet distribution, it is expected that the words only appear in a few of the other topic models. The TF-IDF computation would then just apply a constant factor to all the candidate label scores having no affect on the outcome. A better metric might be to change the IDF computation from including all words that are in the topic to

just the top ten words in other topics.

Using the zero order relevance factor as described in [19], was another way to change the frequency vector. This change had such a detrimental effect to the generated labels that its use was discarded from future experiments.

6.4 Boost Levels

The "linear" boost method shows the most promise. This method simply takes the product of the cosine similarity and the frequency that the label appears in the text. The Culture dataset showed higher scores for the "none" boost method. This method just used the raw cosine similarity metric to rank the labels. I believe this anomaly can be explained by the poor results that the culture dataset showed in general.

6.5 First Order Labeler

As explained in [19], the first order labeler looks at the context of a label to get more information about its semantic meaning. The experiments showed that this information was not helpful in ranking labels, and indeed it lowered the performance of the generated labels. One example from the Computer Science dataset is helpful. For the topic, "protein, sequence, biology, dna, database, gene, bioinformatics, molecular, proteins, sequences", the cosine similarity labeler using the word-topic frequency and linear boost found the best topic was "computational bi-

ology". The first order labeler chose "protein protein". This makes sense because the phrase protein protein interactions are used frequently in computational biology; however, it wouldn't be chosen by the cosine similarity metric because the length of the term vector "protein protein" is 2 and the length of the term vector "computational biology" is square root of 2. Therefore, the cosine similarity metric will be larger for labels that do not repeat words.

6.6 Number of Topics

The "Popular Music" dataset highlights some problems with the number of topics. In the 100 topic dataset, two topics were discovered that shared similar words. In fact, the chosen gold standard categories, and the generated labels, were the same for all three topics.

Topic 1: country,music,texas,bluegrass,nashville,american,songs,folk,western,tennessee
Topic 2: country,music,album,records,american,released,songs,debut,singer,nashville

The chosen gold-standard categories were "country music, american country music, and country music albums". The generated labels were "american country, american country music, and country music". The TF-IDF computation was put in place to look at other topics while generating labels, but it was not effective at doing this. An area of future work, is to add a method to make sure that a topic is discriminative. A good label for topic 2 may be "country music albums", and topic 1 may be "country music styles".

6.7 F Scores

An F measure was used to generate the gold standard. This measure compares the articles that are in the category to the articles that are in the topic. A higher F measure favors narrow categories because it discounts false positives (articles in the topic but not the category). Conversely, a lower F measure favors broad categories because it discounts false negatives (articles in the category but not the topic). Three different F scores were tested, F-0.5, F-1, and F-2. In all three datasets, the F-2 scores were better. The F score were only used when generating the gold standard categories, so this indicates that the experimental labelers were better able to find narrow labels.

Chapter 7

Conclusion

There is much room for improvement, and having a gold standard dataset is a step toward that direction. The methods outlined in this thesis show that a labeler can be automatically evaluated using Wikipedia categories as gold standard labels. Once a good method is obtained, it could be applied outside of the domain of Wikipedia to label other topic models.

Bibliography

[1] K. M. Jackson and W. M. Trochim, "Concept mapping as an alternative approach for the analysis of open-ended survey responses," *Organizational Research Methods*, vol. 5, no. 4, pp. 307–336, 2002.

[2] D. M. Blei and J. D. Lafferty, "Topic models," *Text mining: classification, clustering, and applications*, vol. 10, no. 71, p. 34, 2009.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[4] J. B. Kruskal and M. Wish, *Multidimensional scaling*, vol. 11. Sage, 1978.

[5] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[6] D. Lee, O.-R. Jeong, and S.-g. Lee, "Opinion mining of customer feedback data on the web," in *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pp. 230–235, ACM, 2008.

[7] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger, "Pulse: Mining customer opinions from free text," in *Advances in Intelligent Data Analysis VI*, pp. 121–132, Springer, 2005.

[8] C.-N. Ziegler, M. Skubacz, and M. Viermetz, "Mining and exploring unstructured customer feedback data using language models and treemap visualizations," in *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pp. 932–937, IEEE Computer Society, 2008.

[9] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[10] K. Ganesan, C. Zhai, and J. Han, "Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions," in *Proceedings of the 23rd international conference on computational linguistics*, pp. 340–348, Association for Computational Linguistics, 2010.

[11] E. Riloff, J. Wiebe, and W. Phillips, "Exploiting subjectivity classification to improve information extraction," in *Proceedings of the National Conference On Artificial Intelligence*, vol. 20, p. 1106, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[12] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[13] S. Lee, J. Baker, J. Song, and J. C. Wetherbe, "An empirical comparison of four text mining methods," in *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pp. 1–10, IEEE, 2010.

[14] D. M. Blei and J. D. Lafferty, "A correlated topic model of science," *The Annals of Applied Statistics*, pp. 17–35, 2007.

[15] L. Hong and B. D. Davison, "Empirical study of topic modeling in twitter," in *Proceedings of the first workshop on social media analytics*, pp. 80–88, ACM, 2010.

[16] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, "Topic sentiment mixture: modeling facets and opinions in weblogs," in *Proceedings of the 16th international conference on World Wide Web*, pp. 171–180, ACM, 2007.

[17] W. Xia, Y. He, Y. Tian, Q. Chen, and L. Lin, "Feature expansion for microblogging text based on latent dirichlet allocation with user feature," in *Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International*, vol. 1, pp. 228–232, IEEE, 2011.

[18] Y.-L. Chang and J.-T. Chien, "Latent dirichlet learning for document summarization," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 1689–1692, IEEE, 2009.

[19] Q. Mei, X. Shen, and C. Zhai, "Automatic labeling of multinomial topic models," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 490–499, ACM, 2007.

[20] D. Magatti, S. Calegari, D. Ciucci, and F. Stella, "Automatic labeling of topics," in *2009 Ninth International Conference on Intelligent Systems Design and Applications*, pp. 1227–1232, IEEE, 2009.

[21] J. H. Lau, K. Grieser, D. Newman, and T. Baldwin, "Automatic labelling of topic models," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 1536–1545, Association for Computational Linguistics, 2011.

[22] W. Wang, P. Barnaghi, and A. Bargiela, "Probabilistic topic models for learning terminological ontologies," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 7, pp. 1028–1040, 2010.

[23] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pp. 248–256, Association for Computational Linguistics, 2009.

[24] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, *et al.*, "Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.

[25] S. Bird, "Nltk: the natural language toolkit," in *Proceedings of the COLING/ACL on Interactive presentation sessions*, pp. 69–72, Association for

Computational Linguistics, 2006.

[26] A. K. McCallum, "Mallet: A machine learning for language toolkit." http://mallet.cs.umass.edu, 2002.

Appendix A

DBpedia Example Input

A.1 Long Abstracts

<http://dbpedia.org/resource/Anarchism> <http://dbpedia.org/ontology/abstract> "Anarchism is generally defined as the political philosophy which holds the state to be undesirable, unnecessary, and harmful, or alternatively as opposing authority and hierarchical organization in the conduct of human relations. Proponents of anarchism, known as \" anarchists\", advocate stateless societies based on non-hierarchical voluntary associations. There are many types and traditions of anarchism, not all of which are mutually exclusive. Anarchist schools of thought often differ in fundamental ways, including supporting individualism or collectivism. Schools of anarchist thought are generally divided into the categories of social and individualist anarchism, or similar dual classifications. Anarchism is often considered to be a radical left-wing ideology, and much of anarchist economics and anarchist legal philosophy reflect anti-statist interpretations of communism, collectivism, syndicalism, or participatory economics. Anarchism includes an individualist strain, usually supporting a market economy, private property, and egoism. Some individualist anarchists are also socialists or communists while some anarcho-communists are also individualists. Anarchism as a social movement has regularly endured fluctuations in popularity. The general tendency of anarchism as a social movement has historically been represented by the social anarchist schools anarcho-communism and anarcho-syndicalism, particularly during

its early development. Individualist anarchism has historically been primarily a literary phenomenon, although it had a major impact on the social anarchist thought and individualist anarchists have also participated in large anarchist organizations. Most anarchists oppose all forms of aggression, supporting self-defense or non-violence, while others have supported the use of some coercive measures, including violent revolution and propaganda of the deed, on the path to an anarchist society."@en .
<http://dbpedia.org/resource/Achilles> <http://dbpedia.org/ontology/abstract> "In Greek mythology, Achilles was a Greek hero of the Trojan War, the central character and the greatest warrior of Homer's Iliad. Plato named Achilles the most handsome of the heroes assembled against Troy. Later legends (beginning with a poem by Statius in the 1st century AD) state that Achilles was invulnerable in all of his body except for his heel. As he died because of a small wound on his heel, the term Achilles' heel has come to mean a person's principal weakness."@en .
<http://dbpedia.org/resource/A> <http://dbpedia.org/ontology/abstract> "A is the first letter and a vowel in the ISO basic Latin alphabet. It is similar to the Ancient Greek letter Alpha, from which it derives."@en .
<http://dbpedia.org/resource/Albedo> <http://dbpedia.org/ontology/abstract> "Albedo, or reflection coefficient, derived from Latin albedo \"whiteness\" (or reflected sunlight), in turn from albus \"white\", is the diffuse reflectivity or reflecting power of a surface. It is defined as the ratio of reflected radiation from the surface to incident radiation upon it. Being a dimensionless fraction, it may also be expressed as a percentage, and is measured on a scale from zero for no reflecting power of a perfectly black surface, to 1 for perfect reflection of a white surface. Albedo depends on the frequency of the radiation. When quoted unqualified, it usually refers to some appropriate average across the spectrum of visible light. In general, the albedo depends on the directional distribution of incoming radiation. Exceptions are Lambertian surfaces, which scatter radiation in all directions according to a cosine function, so their albedo does not depend on the incident distribution. In practice, a bidirectional reflectance distribution function (BRDF) may be required to characterize the scattering properties of a surface accurately, although the albedo is a very useful first approximation. The albedo is an important concept in climatology and astronomy, as well as in calculating reflectivity of surfaces in LEED sustainable rating systems for buildings, computer graphics and computer vision. The

```
    average overall albedo of Earth, its planetary albedo, is 30
    to 35%, because of the covering by clouds, but varies widely
    locally across the surface, depending on the geological and
    environmental features. The term was introduced into optics by
     Johann Heinrich Lambert in his 1760 work Photometria."@en .
```

A.2 Article Titles

```
<http://dbpedia.org/resource/Alain_Connes> <http://www.w3.org
    /2000/01/rdf-schema#label> "Alain Connes"@en .
<http://dbpedia.org/resource/Allan_Dwan> <http://www.w3.org
    /2000/01/rdf-schema#label> "Allan Dwan"@en .
<http://dbpedia.org/resource/Aircraft_Carrier> <http://www.w3.org
    /2000/01/rdf-schema#label> "Aircraft Carrier"@en .
<http://dbpedia.org/resource/Actress> <http://www.w3.org/2000/01/
    rdf-schema#label> "Actress"@en .
<http://dbpedia.org/resource/List_of_Atlas_Shrugged_characters> <
    http://www.w3.org/2000/01/rdf-schema#label> "List of Atlas
    Shrugged characters"@en .
<http://dbpedia.org/resource/America_the_Beautiful> <http://www.
    w3.org/2000/01/rdf-schema#label> "America the Beautiful"@en .
<http://dbpedia.org/resource/Ayn_Rand> <http://www.w3.org
    /2000/01/rdf-schema#label> "Ayn Rand"@en .
<http://dbpedia.org/resource/Alchemy> <http://www.w3.org/2000/01/
    rdf-schema#label> "Alchemy"@en .
<http://dbpedia.org/resource/Anthropology> <http://www.w3.org
    /2000/01/rdf-schema#label> "Anthropology"@en .
<http://dbpedia.org/resource/Abacus> <http://www.w3.org/2000/01/
    rdf-schema#label> "Abacus"@en .
<http://dbpedia.org/resource/Air_Transport> <http://www.w3.org
    /2000/01/rdf-schema#label> "Air Transport"@en .
```

A.3 Article Redirects

```
<http://dbpedia.org/resource/Closest_pair> <http://dbpedia.org/
    ontology/wikiPageRedirects> <http://dbpedia.org/resource/
    Closest_pair_of_points_problem> .
<http://dbpedia.org/resource/Silver_Stars_(South_Africa)> <http
    ://dbpedia.org/ontology/wikiPageRedirects> <http://dbpedia.org
    /resource/Platinum_Stars_F.C.> .
<http://dbpedia.org/resource/Kyeong-yeong_Lee> <http://dbpedia.
    org/ontology/wikiPageRedirects> <http://dbpedia.org/resource/
    Lee_Geung-young> .
<http://dbpedia.org/resource/Vichy_Springs> <http://dbpedia.org/
    ontology/wikiPageRedirects> <http://dbpedia.org/resource/
    Vichy_Springs,_California> .
```

<http://dbpedia.org/resource/Culoptila_tarascanica> <http://dbpedia.org/ontology/wikiPageRedirects> <http://dbpedia.org/resource/Culoptila> .
<http://dbpedia.org/resource/Template:Cite_pmid/21359919> <http://dbpedia.org/ontology/wikiPageRedirects> <http://dbpedia.org/resource/Template:Cite_doi/10.1007.2Fs11655-011-0665-7> .
<http://dbpedia.org/resource/Carmen_Bristoliense> <http://dbpedia.org/ontology/wikiPageRedirects> <http://dbpedia.org/resource/Bristol_Grammar_School> .
<http://dbpedia.org/resource/Buenos_Aires_Underground> <http://dbpedia.org/ontology/wikiPageRedirects> <http://dbpedia.org/resource/Buenos_Aires_Metro> .
<http://dbpedia.org/resource/J._w._s._cassels> <http://dbpedia.org/ontology/wikiPageRedirects> <http://dbpedia.org/resource/J._W._S._Cassels> .

A.4 Category Titles

<http://dbpedia.org/resource/Category:Futurama> <http://www.w3.org/2000/01/rdf-schema#label> "Futurama"@en .
<http://dbpedia.org/resource/Category:World_War_II> <http://www.w3.org/2000/01/rdf-schema#label> "World War II"@en .
<http://dbpedia.org/resource/Category:Professional_wrestling> <http://www.w3.org/2000/01/rdf-schema#label> "Professional wrestling"@en .
<http://dbpedia.org/resource/Category:Programming_languages> <http://www.w3.org/2000/01/rdf-schema#label> "Programming languages"@en .
<http://dbpedia.org/resource/Category:Algebra> <http://www.w3.org/2000/01/rdf-schema#label> "Algebra"@en .
<http://dbpedia.org/resource/Category:Anime> <http://www.w3.org/2000/01/rdf-schema#label> "Anime"@en .
<http://dbpedia.org/resource/Category:Abstract_algebra> <http://www.w3.org/2000/01/rdf-schema#label> "Abstract algebra"@en .
<http://dbpedia.org/resource/Category:Linear_algebra> <http://www.w3.org/2000/01/rdf-schema#label> "Linear algebra"@en .
<http://dbpedia.org/resource/Category:Mathematics> <http://www.w3.org/2000/01/rdf-schema#label> "Mathematics"@en .
<http://dbpedia.org/resource/Category:Monarchs> <http://www.w3.org/2000/01/rdf-schema#label> "Monarchs"@en .

A.5 Article Categories

<http://dbpedia.org/resource/Autism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Autism> .
<http://dbpedia.org/resource/Autism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:

Communication_disorders> .
<http://dbpedia.org/resource/Autism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Mental_and_behavioural_disorders> .
<http://dbpedia.org/resource/Autism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Neurological_disorders> .
<http://dbpedia.org/resource/Autism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Neurological_disorders_in_children> .
<http://dbpedia.org/resource/Autism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Pervasive_developmental_disorders> .
<http://dbpedia.org/resource/Autism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Psychiatric_diagnosis> .
<http://dbpedia.org/resource/Autism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Learning_disabilities> .
<http://dbpedia.org/resource/Anarchism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Anarchism> .
<http://dbpedia.org/resource/Anarchism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Political_culture> .
<http://dbpedia.org/resource/Anarchism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Political_ideologies> .
<http://dbpedia.org/resource/Anarchism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Social_theories> .
<http://dbpedia.org/resource/Anarchism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Anti-fascism> .
<http://dbpedia.org/resource/Anarchism> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Greek_loanwords> .
<http://dbpedia.org/resource/Agricultural_science> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Agronomy> .
<http://dbpedia.org/resource/Albedo> <http://purl.org/dc/terms/subject> <http://dbpedia.org/resource/Category:Climate_forcing> .

A.6 Category Relationships

<http://dbpedia.org/resource/Category:Middle-earth_languages> <http://www.w3.org/2004/02/skos/core#broader> <http://dbpedia.

org/resource/Category:Artistic_languages> .
<http://dbpedia.org/resource/Category:Chess> <http://www.w3.org/2004/02/skos/core#broader> <http://dbpedia.org/resource/Category:Mind_sports> .
<http://dbpedia.org/resource/Category:Philosophers> <http://www.w3.org/2004/02/skos/core#broader> <http://dbpedia.org/resource/Category:Philosophy> .
<http://dbpedia.org/resource/Category:Cereals> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/2004/02/skos/core#Concept> .
<http://dbpedia.org/resource/Category:Ordinary_differential_equations> <http://www.w3.org/2004/02/skos/core#prefLabel> "Ordinary differential equations"@en .
<http://dbpedia.org/resource/Category:Snooker> <http://www.w3.org/2004/02/skos/core#broader> <http://dbpedia.org/resource/Category:Precision_sports> .
<http://dbpedia.org/resource/Category:Oceania> <http://www.w3.org/2004/02/skos/core#prefLabel> "Oceania"@en .
<http://dbpedia.org/resource/Category:South_African_people> <http://www.w3.org/2004/02/skos/core#prefLabel> "South African people"@en .

Appendix B

Stopwords

a
able
about
above
according
accordingly
across
actually
after
afterwards
again
against
all
allow
allows
almost
alone
along
already
also
although
always
am
among
amongst
an
and
another
any
anybody
anyhow
anyone
anything
anyway
anyways
anywhere
apart
appear
appreciate
appropriate
are
around
as
aside
ask
asking
associated
at
available
away
awfully
b
be
became
because
become
becomes
becoming
been
before
beforehand
behind
being
believe
below
beside
besides
best
better
between
beyond
both
brief
but
by
c
came
can
cannot
cant
cause
causes
certain
certainly
changes
clearly
co
com
come
comes
concerning
consequently
consider
considering
contain
containing
contains
corresponding
could
course
currently
d
definitely
described
despite
did
different
do
does
doing
done
down
downwards
during
e
each
edu
eg
eight
either
else
elsewhere
enough
entirely
especially
et
etc
even
ever
every
everybody
everyone
everything
everywhere
ex
exactly
example
except
f
far
few
fifth
first
five
followed
following
follows
for
former
formerly
forth
four
from
further
furthermore
g
get
gets
getting
given
gives
go
goes
going
gone
got
gotten
greetings
h
had

happens
hardly
has
have
having
he
hello
help
hence
her
here
hereafter
hereby
herein
hereupon
hers
herself
hi
him
himself
his
hither
hopefully
how
howbeit
however
i
ie
if
ignored
immediate
in
inasmuch
inc
indeed
indicate
indicated
indicates
inner
insofar
instead
into
inward
is
it
its
itself
j
just
k
keep
keeps
kept
know
knows
known
l
last
lately
later
latter
latterly
least
less
lest
let
like
liked
likely
little
look
looking
looks
ltd
m
mainly
many
may
maybe
me
mean
meanwhile
merely
might
more
moreover
most
mostly
much
must
my
myself
n
name
namely
nd
near
nearly
necessary
need
needs
neither
never
nevertheless
new
next
nine
no
nobody
non
none
noone
nor
normally
not
nothing
novel
now
nowhere
o
obviously
of
off
often
oh
ok
okay
old
on
once
one
ones
only
onto
or
other
others
otherwise
ought
our
ours
ourselves
out
outside
over
overall
own
p
particular
particularly
per
perhaps
placed
please
plus
possible
presumably
probably
provides
q
que
quite
qv
r
rather
rd
re
really
reasonably
regarding
regardless
regards
relatively
respectively
right
s
said
same
saw
say
saying
says
second
secondly
see
seeing
seem
seemed
seeming
seems
seen
self
selves
sensible
sent
serious
seriously
seven
several
shall
she
should
since
six
so
some
somebody
somehow
someone
something
sometime
sometimes
somewhat
somewhere
soon
sorry
specified
specify
specifying
still
sub
such
sup
sure
t
take
taken
tell
tends
th
than
thank
thanks
thanx
that

thats
the
their
theirs
them
themselves
then
thence
there
thereafter
thereby
therefore
therein
theres
thereupon
these
they
think
third
this
thorough
thoroughly
those
though
three
through
throughout
thru
thus
to
together
too
took
toward
towards
tried
tries
truly
try
trying
twice
two
u
un
under
unfortunately
unless
unlikely
until
unto
up
upon
us
use
used
useful
uses
using
usually
uucp
v
value
various
very
via
viz
vs
w
want
wants
was
way
we
welcome
well
went
were
what
whatever
when
whence
whenever
where
whereafter
whereas
whereby
wherein
whereupon
wherever
whether
which
while
whither
who
whoever
whole
whom
whose
why
will
willing
wish
with
within
without
wonder
would
x
y
yes
yet
you
your
yours
yourself
yourselves
z
zero
’s
’t
’d