Topic Modelling on Novels of Different Lengths Using Latent Dirichlet Allocation

Alec Situ *University Hill Secondary School*
MehtA+ Tutoring

Daniel Suh *Orange High School*
MehtA+ Tutoring

Justin Wickelgren *LASA High School*
MehtA+ Tutoring

Editor: Alec Situ

Abstract

By performing topical analysis models on fictional novels such as *Sherlock Holmes*, *Harry Potter*, *To Kill a Mocking Bird*, and *The Book of Negroes*, we discovered multiple topics and words in the book associated with the topics. As we optimize our model, machines will be able to recognize topics many humans could do, and perhaps even expand upon human knowledge on topics and connections within certain stories. Thus, we explored using state-of-the-art TF-IDF model in conjunction with Latent Dirichlet Allocation (LDA) on novels of varying lengths in attempt to 1. affirm in novels of what length can the model identify topics accurately and 2. optimize the number of topics so the model can produce ideal results. Although we tried many other models, including normal LDA, LDAMulticore, LDAMallet, and Hierarchical Dirichlet Process (HDP), we found that our model with TF-IDF generated results dominating that of other models.

Keywords: Latent Dirichlet Allocation, TF-IDF, topical analysis, novel analysis

1. Introduction

As of right now, humans can comprehend books much more rigorously than machines can, even if machines are well-trained and optimized. Conversely, machines can read books significantly faster and discover results and connections in books humans cannot. Thus, it

©2021 Alec Situ and Daniel Suh and Justin Wickelgren.

may be advantageous to create models which can effectively and efficiently produce relevant topics of a book. This method may be particularly useful when analyzing newly published books or obscure, less well-known books. Moreover, we'd like to discover the underlying and taciturn relationships and topics within the book.

Extracting topics from books is a novel, yet growing field of machine learning. To expand human knowledge when creating and analyzing novels, models will need to be improved. Our model tackles the many unresolved questions of topic modeling using the availability of novels we have. We used different preprocessing methods, different models, and different books to try which models work the best for the proposed question.

2. Related Works

One of our inspirational research papers was Stanford Literary Lab's *Pamphlet 10*, published in October 2015 on paragraphs as an area of focus for topical analysis. Algee-Hewett et al. (2015). Through this article, we were able to obtain a thorough understanding of how to use paragraphs as documents in our model, such as the number of topics and themes present in each paragraph, and ways to deal with dialogue paragraphs. We tested each of these methods to see which ones in combination will work the best.

Another paper we based our modelling upon was an entry in ICSMM 2018 which described their attempts at using TF-IDF with LDA models. As we found their results promising, we did the same, and we ended up with results significantly better than those of normal LDA models such as (LDA, LDAMulticore, and LDAMallet).

3. Methodology

3.1 Books

Our selection of books included:

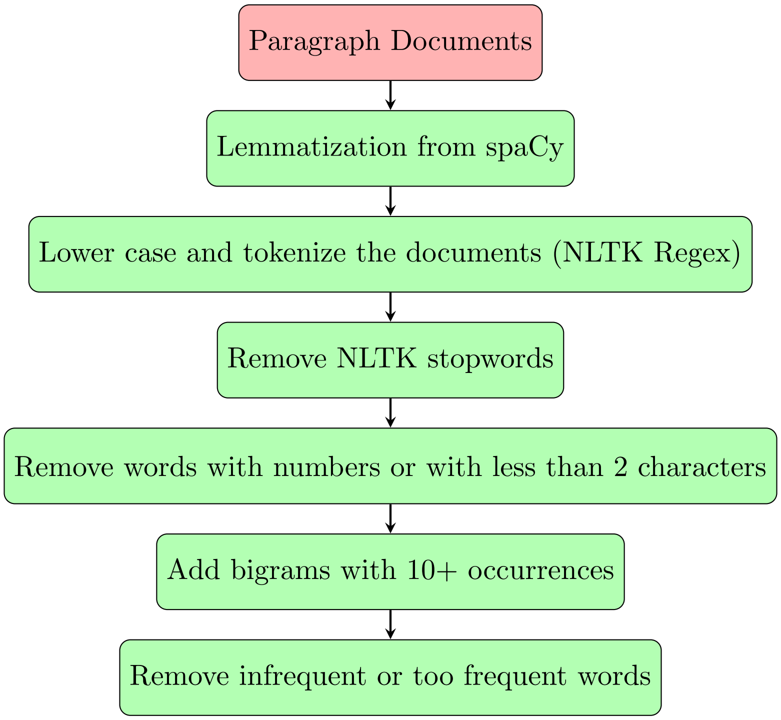
- *To Kill a Mocking Bird* (281 Pages)
- *The Book of Negroes* (511 Pages)
- *Sherlock Holmes* (Whole Series or 1096 Pages)
- *Harry Potter* (Whole Series or 4224 Pages)

Evidently, the books had different lengths, and we tested our model on all of the books.

3.2 Preprocessing

We used relatively straightforward preprocessing methods. We started out by getting rid of unnecessary texts and strings in the raw texts. In the aforementioned section, we said that we split the text by paragraphs. For poems such as *The Book of Negroes* and *Sherlock Holmes*, we tried concatenating shorter paragraphs (paragraphs of less than 30 words) so each document can have a decent length. Although it can be disputed whether this was beneficial or not, we found that this was not inimical.

Then, we did the following:


Paragraph Documents
Lemmatization from spaCy
Lower case and tokenize the documents (NLTK Regex)
Remove NLTK stopwords
Remove words with numbers or with less than 2 characters
Add bigrams with 10+ occurrences
Remove infrequent or too frequent words

It is notable that we tried a few other preprocessing methods. For example, we tried using NLTK's lemmatizer, but it would only lemmatize nouns. Although we attempted to lemmatize all the significant parts of speech, errors and time constraints prevented us from doing so. So we tried to use spaCy's lemmatizer, even though the results stayed relatively same.

These steps of preprocessing leads us to what we have in the model.

3.3 Number of Topics

The optimal number of topics could be found using the Jaccard index, or Jaccard similarity, in conjunction with topic coherence. The Jaccard index and topic coherence can be defined as follows:

$$J(A,B) = \frac{A \cup B}{A \cap B} \tag{1}$$

$$coherence = \sum_{i<j} score(w_i, w_j) \quad \text{where} \quad score(w_i, w_j) = \log(\frac{p(w_i, w_j)}{p(w_i)p(w_j)}) \tag{2}$$

$p(w_i)$ can be defined as the probability that w_i is found in a document, and $p(w_i, w_j)$ can be defined as the probability that both w_i and w_j can be found in the same document.

The result of the Jaccard index is a value between 0 and 1, in which a value closer to 0 means that the 2 sets are further apart, while a value closer to 1 means that the 2 sets are closer together. The result of the coherence score is also a value between 0 and 1, in which a value closer to 1 means the words in any given topic are more related to each other.

We wanted to maximize the topic coherence, and minimize the Jaccard index. By doing this, the words in each topic would be more closely related, while also achieving our goal of having unique and distinct words in each topic. See Figure (2) for the graph of both of these values.

3.4 Modelling

For almost all of the modelling, we used the Gensim library. Initially, we tried the most basic version: a standard LDAModel with a Bag of Words (BoW) corpus. The results were bad, which was to be expected. A few trials later, we discovered that a TF-IDF corpus was far superior than the BoW corpus. Thus, for all the models, we used a TF-IDF corpus. Other vectorizations we thought about was Word2Vec and Doc2Vec, however time constraints prevented us from testing them.

There were also different models in which we tested our corpus on. Gensim has many LDA models, such as LDAModel, LDAMulticore, and LDAMallet. (Note that LDAMallet is incompatible with TF-IDF corpus, so we didn't use it.) Amongst non-LDA models, there is Latent Semantic Analysis (LSA) and Hierarchical Dirichlet Process (HDP). However, neither of those two models gave significantly better results than ones given by LDA, so we discarded them. Therefore, we used either LDAModel or LDAMulticore, which gave similar results.

4. Results

Conspicuously, the results improved as the length of the book increased, because there were more documents and a larger dataset to train on. The results for *To Kill a Mockingbird* were relatively weak. However, as we start to move towards longer novels or series (*The Book of Negroes*, *Sherlock Holmes*, and *Harry Potter*), the results become clearer and more coherent; we were able to identify topics and certain keywords that are vital to the story. Furthermore, the model was able to identify topics which, although not vital to the books, were nonetheless interesting. However, the model only outputted words that the model deemed connected with the topic, not the topic itself.

In Figure 1 is an image of the topic similarity (in 8 topics):

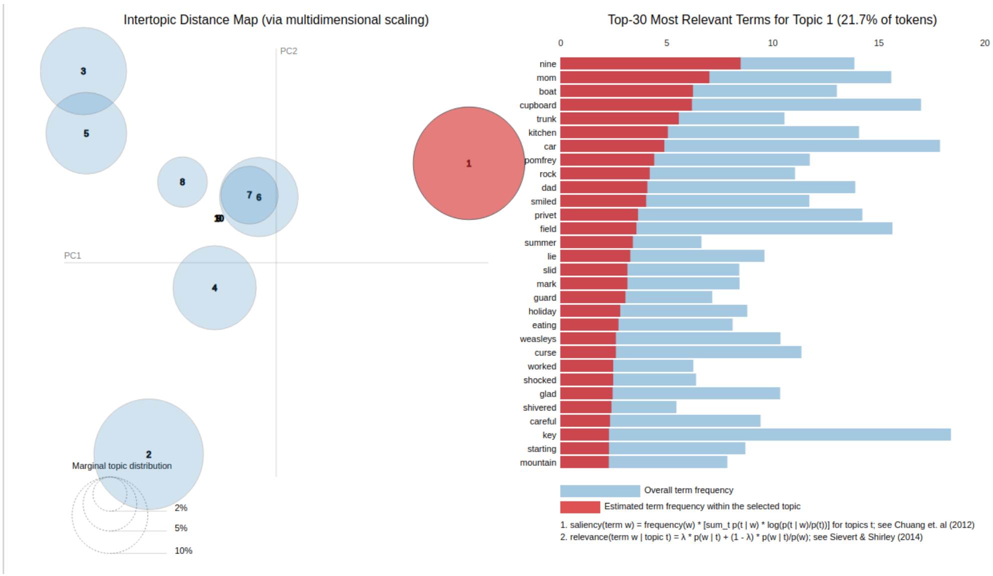


Figure 1: Topic similarity in Harry Potter

As to answer the question regarding the optimal number of topics for the model, we have Figure 2:

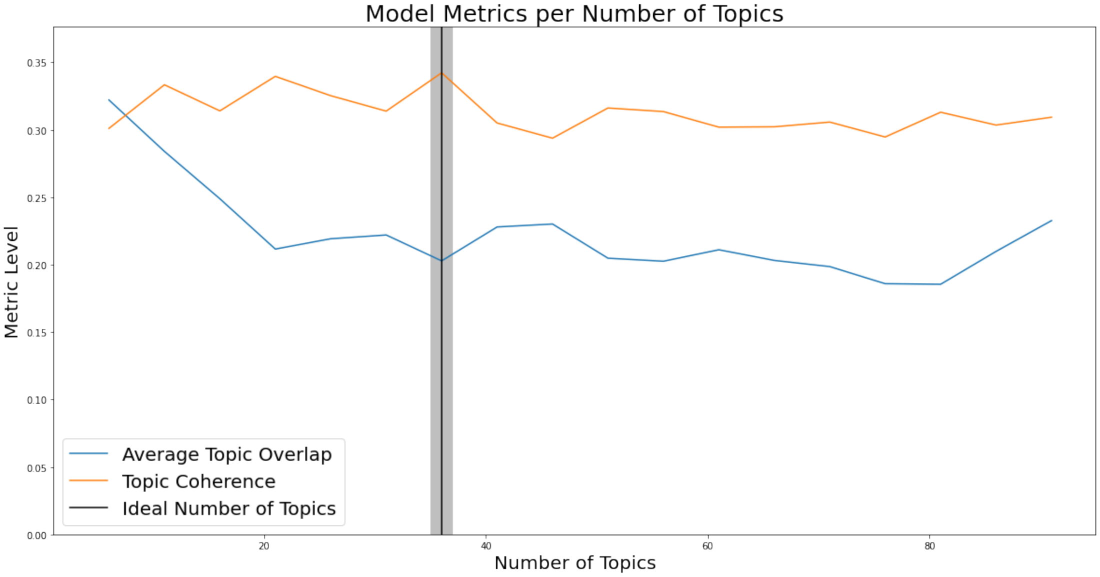


Figure 2: Optimal number of topics for Harry Potter

The graph discerns the topic coherence and topic overlap (orange line is coherence and blue line is overlap), or in other words, Jaccard Similarity. We would like to minimize overlap while maximizing coherence. From the visualization, it is clear that the optimal number of topics is roughly 35 topics.

5. Conclusion and Future Works

Our LDA models vary a little bit for every book, because the preprocessing is different, the content is different, and most importantly, the amount of training the model gets is different. Thus, we see that identifying topics can be very hard, particularly when we only get the words associated with the topic, not the topic itself. Moreover, the results were less than we had hoped with some words outputted being overly common/obvious words that had no effect on certain topics. As our model improves, so will the results. In the future, we would like to optimize our LDA model by

1. optimizing the parameters for the model
2. create better preprocessing techniques (particularly the lemmatizer) to improve the quality of our documents
3. further transformations with Word2Vec and Doc2Vec
4. or even improve other "failed" models, such as HDP

All of this will contribute to a better understanding of how machines can help humans analyze texts and identify topics.

Acknowledgments

We would like to thank Professor Elkins and Professor Chun of Kenyon College. We also thank the instructors at MehtA+. Particularly, our main instructor: Bhagirath Mehta.

References

Mark Algee-Hewett, Ryan Heuser, and Franco Moretti. On paragraphs. scale, themes, and narrative forms. *Stanford Literary Lab*, 10, 2015.