AUTHOR AND WRITING STYLE IDENTIFICATION

Yumin Zhang, Hantao Zhao

yumzhang@umich.edu, hantaoz@umich.edu

University of Michigan - School of Information

Introduction

Author identification, which is widely utilized in finding ghost writer and detecting plagiarism, has long been popular topics in Natural Language Processing(NLP). In this project, we build models using the documents and their authors, and train the model to predict the most likely author of a input query. In our project, we gained in-depth experience of building complicated models for information retrieval tasks. We have built Long short-term memory(LSTM), BM25, Latent Semantic Indexing(LSI) and Latent Dirichlet Allocation(LDA) models in our project. These models have different performances, and each of them are significantly better than the baseline model. We have learned that some models might be more competitive in performing our task. In our tests, an average accuracy higher than 90% has been reached.

Data Source & Pre-processing

The dataset used for this project is *Fifty Victorian Era Novelists Authorship Attribution Data*, which comes from UC Irvine Machine Learning Repository. The data are extracted from GDELT database. The dataset can be downloaded here. All the authors in the dataset are filtered through the following criteria: English Writing Authors, have enough books available (at least 5) and are 19th century authors. The corpus is built with books collected from these authors, and the building process of the dataset undergoes the following processes. First, all books have heads and tails removed. Next, the top 10,000 words are selected from the corpus, and the other words are removed while keeping the rest sentence structure. Finally, all books are split into text fragments, with each has length of 1,000 words as an individual sample in the dataset, which is how the dataset is formed.

This dataset contains texts from 45 different authors, with 53679 rows of data. The data format is in 2 columns. The first column is text, contains text of 1,000 words, and the second column is author, represents the id of author.

Methods

We perform preprocessing including lowercase transformation and stopwords removal. Exploration on dataset displays some suggestive information, such as distribution of word count in document, which is shown in Fig. 1. We split the whole dataset into test set, and training set including 80% of the items. We select LSTM as our baseline model, and compare performance of LDA, LSI and BM25 to determine the most suitable one. Some of the key models and their results are shown in the next part.

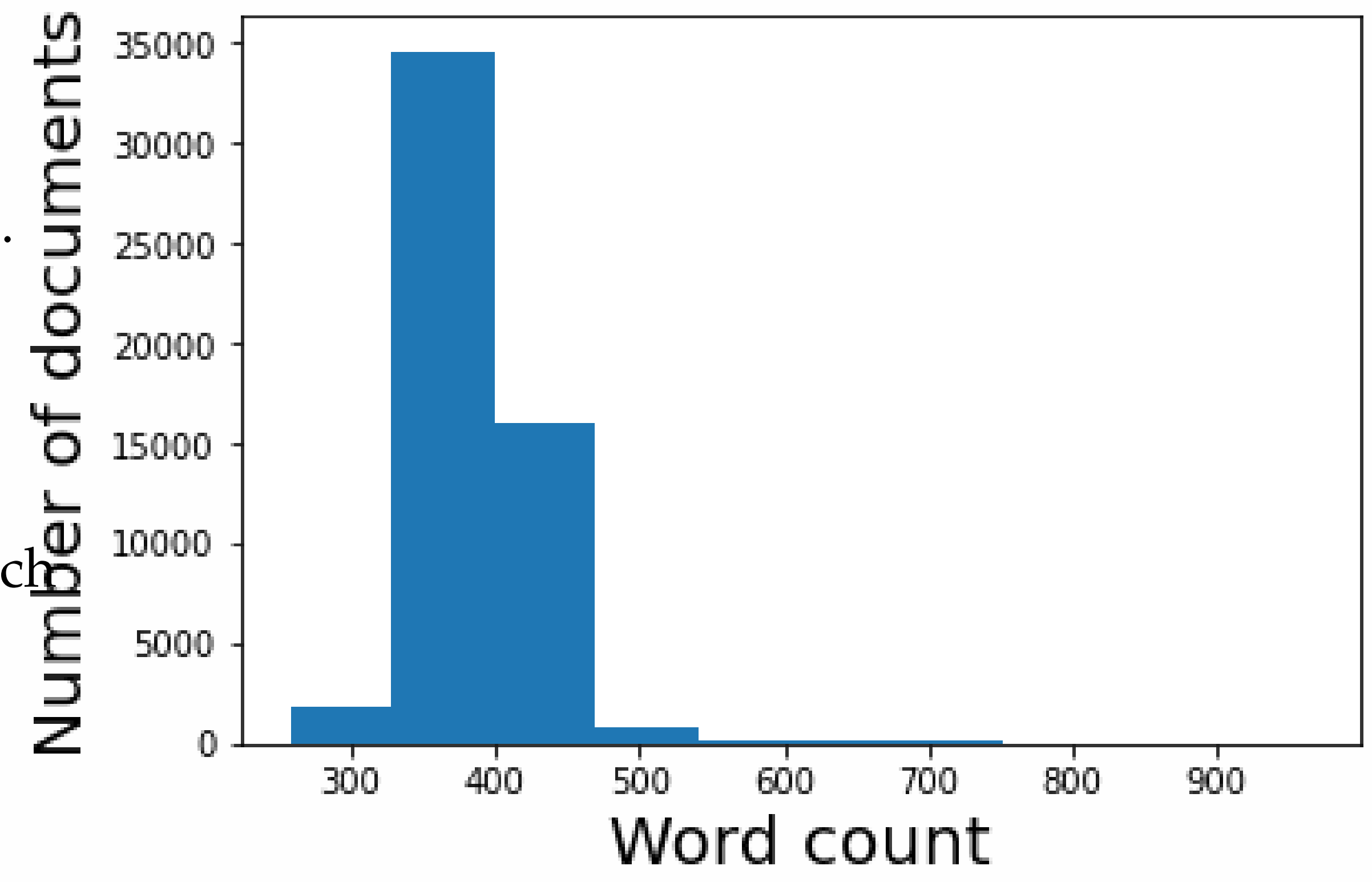

Figure 1: Word Count Distribution

Evaluation and Results

LDA

We decide to introduce NLP models including LDA. Our motivation for the implementation is that they can reveal the underlying information of documents. From the results in homework2, LDA does perform better than BM25. It takes synonyms, context and underlying structures of sentences into consideration. By using LDA, we can analyze the meanings of sentences and analyze their similarity in depth. We determine the number of topics in LDA to be equal to 200. We perform experiments to test relation between model performance and max sentence length, which results are shown in Fig. 2.

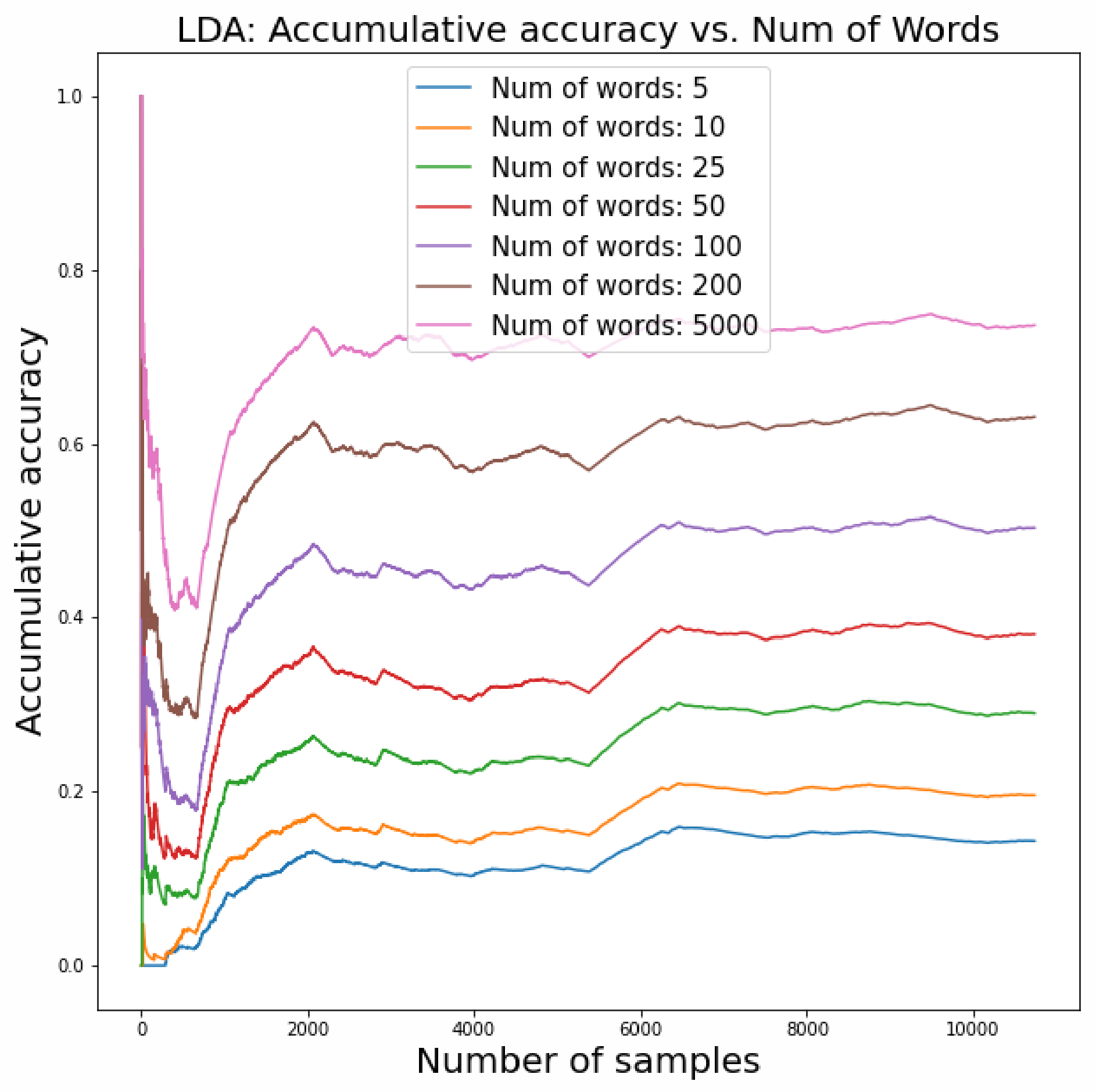

Figure 2: LDA Performance

BM25

For the bm25 method, we set our model to be metapy.index.OkapiBM25(k1 = 1.2, b = 0.5, k3 = 500).

Since the parameters of our BM25 model has been fixed, the most important parameter in our program is the number of words in the query. As mentioned earlier, each document contains about one thousand of words. If the entire document is used as the query, it would lead to incredible high accuracy of about 95%. In the accumulative accuracy plot below, this corresponds to the result of the top curve, where setting the number of words to be 5000 means that the entire document is used as the query. We can see that its accuracy is stably above 90%. Inputting the entire document, however, is literally too strict. We try to reduce the number of words, and the accuracy starts to drop. The overall accuracy of BM25 model versus the number of words is listed in the table.

Number of words	Avg. accuracy for bm25
5	0.30763493987135265
10	0.5268747088961342
25	0.6614811364694924
50	0.7578241430700448
100	0.8441691505216096
200	0.911978390461997
5000	0.955570044709389

Table 1: Num of words vs. Average accuracy for BM25

Conclusions and Discussions

We can see that LDA can achieve performance of 73.6%, while BM25 can achieve performance of 95.6%, which highly perform the former, and is selected as our final model. Compared to LDA which transforms documents into vectors and calculate the nearest, BM25 directly leads query to the highest ranking items, which highlights more on the similarity between documents, which accounts for its advantage. BM25 model well qualifies for the task of author identification.

What's Next

Some reference mention that LSI and LDA models could have good performances when the query words can be found in topics. This, however, requires that only the most representative words from a document is put into the query. We wonder how we can select these most representative words from a given document. We have noticed that the documents of some authors are easier to be identified in all of the models. If we had enough time, we could take a deeper look at the underlying reason for this.

References

[1] UCI. 2018. Victorian Era Authorship Attribution Data Set