Automated Phrase Mining from Massive Text Corpora
DSC 180A - Final Report

Yicen Ma, Tiange Wan, Anant Gandhi

March 07, 2021

1 Abstract

We propose the creation of a full-stack website as an extension of the AutoPhrase algorithm and text analysis to help the non-tech users understand their text efficiently. Also, we provide a notebook with one specific data set as an example to the users.

2 Introduction

Phrase mining - the extraction of high-quality phrases from a large input corpus - is an important and interesting skill to analyze text data sets with. It identifies the phrases instead of an uni-gram word, which provides a much more understanding of the text. For instance, "support vector machine" should be defined as a high-quality phrase instead in machine learning scientific writing. There is a strong connection between uni-gram "support", "vector" and "machine" and it is much more meaningful than any of those uni-grams with high frequency of the phrase appearing in the writing (Jialu 1). Phrase mining has various applications such as information extraction, topic modeling, etc. It also plays an important role in domains like market research, public sentiment and fraud detection. Raw frequency-based phrase mining has many limitations, most majorly that recurring word sequences may not form meaningful phrases (or be structured), which causes semantic ambiguity and misleading quality assessment. The purpose of this project is to help non-tech users understand their text efficiently by utilizing AuoPhrase algorithms and other text analysis techniques and produce the results by concise steps. AutoPhrase has a better performance in mining phrases from a large corpus with minimizing human labeling effort with domain independence in any provided knowledge base languages. AutoPhrase is useful, interesting and well-structured for real-world applications. But it is not directly helpful for non-tech users/common people in implementing. Therefore, in the website development, we generate 3 front webs and one back-end on running

AutoPhrase and producing results. The users only need to upload their files and the process will run automatically to generate word clouds and the phrase text with the weight values. In the notebook, there is more text analysis in the forms of texts and charts to help them understand the text visually. Our project mainly includes:

- A web application, which requires the user to upload their own input file (in txt format) and automatically produce the results
- A visualization and dynamic dashboard for relevant outputs on the website
- Available to append their knowledge base the default knowledge base on the website

3 Related Works

In the previous quarter, we understood the algorithm behind the AutoPhrase and created an application of it. However, it is hard for a non-technical person to apply this method into their real life, because they need to implement some code in the terminal, writing the bash files, or python code, etc., which is not helpful in understanding the text. Furthermore, the final output of the AutoPhrase only generates the phrases with descending values, which will confuse the users. Therefore, we decided to generate a web to solve the above problems. To make more people understand AutoPhrase and their text within data science techniques, we decided to build a web visualization and add more other text analysis techniques to generate charts, word-cloud, text presentation with explanations for the results. The users can upload their own text, and see the progress of the AutoPhrase steps on the screen. Lastly, all of the visualizations and text results are shown in front of them.

4 Methods

1. AutoPhrase (Automated Phrase Mining):

Most of the phrase mining methods are expensive and depend on a lot of human labeling efforts to train the model in a specific domain and language. However, AutoPhrase has a better performance in mining phrases from a large corpus with minimizing human labeling effort with domain independence in any provided knowledge base languages(Jingbo 1). It performs based on Robust Positive-Only Distant Training and POS-Guided Phrasal Segmentation methods to compute the values of phrase quality.

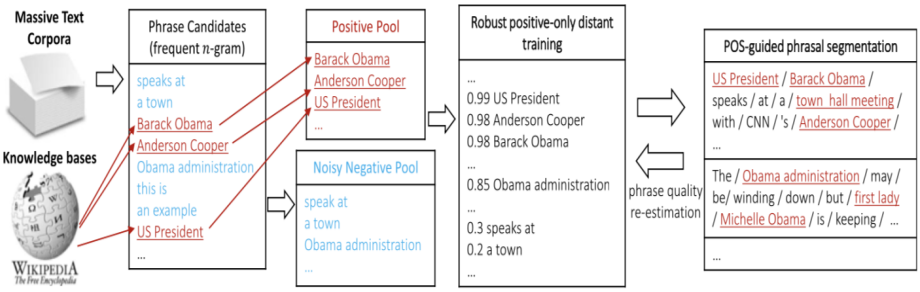

Figure 1: Outline of Automated Phrase Mining using Robust positive only distant training and POS-guide phrasal segmentation

AutoPhrase method requires inputs as text corpus sequence of words in one language and a knowledge base file and gets the ranked quality value of phrases in descending order as outputs. The knowledge base is a file of the term database, which used to distinguish the common terms. In this method, the default knowledge base is from the phrases on Wikipedia. The AutoPhrase method contains data cleaning steps in the method as data processing. For the further steps, it uses the phase mining methods including tokenizing and POS-tagging. It splits the text input into a smaller unit and to mark up the words in the text to prepare the phrases. Later, it applies phrasal segmentation to break a sequence/sentence into a semantic unit (a word/phrase). To compute the quality of candidate phrases from a raw corpus, AutoPhrase depends on knowledge bases to create a clean and free positive phrase pool in distance supervision to replace experts provided labels. Then, it implements the noisy reduction in the noisy negative pool and applies the POS-Guided phrasal segmentation to meet the requirement of high-quality phrases. At the end, it produces outputs of ranked lists with the decreasing quality phrases.

2. Web Development:

In the front web development, we built three front webs. We utilize hyper text markup language (HTML) to structure and design in a web browser with the assistance of Cascading style sheeting (CSS), which is the style presentation and JavaScript (JS) enabled to interactive web pages. Furthermore, we utilized LayUI as a front-end UI framework to make it more organized. It is usable for users to upload their own files, the knowledge base, reading the processing steps of AutoPhrase and some text and word cloud visualization at the end.
In the back-end web development, we build the back-end based on the Apache Tomcat, which is the HTTP web server environment and write the code in JavaScript, Java and Python. To accomplish the implementation and configuration, we used Java Development Kit and Spring MVC and Spring as the

application framework to control the back-end of the web. It will read the input file/knowledge base and stored into the web folder; call the bash file to start the AutoPhrase processing and the generate other analysis results; read the text's result and graphs; store those into the web folder; and produce the top 20 valuable phrases with corresponding size in the word cloud.

3. Term Frequency/Inverse Document Frequency(TF-IDF) Analysis:

TF-IDF evaluates how relevant the word is in one document as an information retrieval method. It works by increasing proportionally to the number of times a word appears in a document, but is offset by the number of documents that contain the word (MonkeyLearn 4). It is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. For example, the stop words such as "that", "with", "to" will be ignored, because those words are meaningless to represent the whole corpus as a whole. This method is often used as a weighting factor in searches of information retrieval, text mining, and user modeling.
The calculation is as follows:

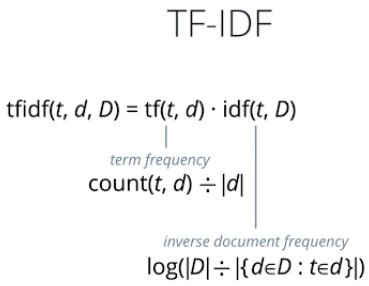

Figure 2: TF-IDF Formula

We use the TfidfVectorizer class from the sklearn.feature_extraction.text library to perform TF-IDF analysis and combine it with the AutoPhrase to generate the word with corresponding quality value on the analysis. Sentiment Analysis It is one of the uses of Natural Language Processing (NLP) to study the subject information and affective states from text. Sentiment analysis (or opinion mining) is a NLP technique used to determine whether data is positive, negative or neutral. We use a Python library called text blob to analyze both a polarity and subjectivity score on each sentence of the text corpus. Then plot a distribution based on that. This would give users an insight on whether the corpus is positive or negative overall and if it's more rational or emotional in the text.

5 Results

On the first web page, it shows a brief introduction of the methods and usages of the web. The users can upload their own input text or knowledge base by clicking on the upload buttons. After uploading, the text presentation will automatically be updated as presentations. They can also upload their own knowledge base by appending to the default file. Later, they can click the button "Run AutoPhrase" to start the analysis process. In this demo, we used one data set called "DBLP.5k", which is the first 5000 lines on the "DBLP" file and it contains the topical key phrases from the scientific paper domain. We use this data set, because it can generate results faster than the whole DBLP data set, which is about 10 mins.

Figure 3: Screenshot of the first web page

On the second web page, it will present the progress of the running process step by step. The whole process contains data cleaning, exploratory data analysis, AutoPhrase analysis, and generating the graphs and images.

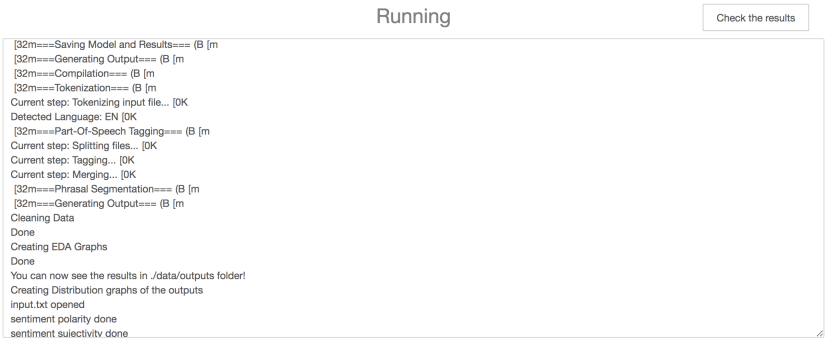

Figure 4: Screenshot of the second web page

On the third web page, it presents the text and visualization results corresponding to their visualization, which all the results are shown in the below section.

Text Analysis & Visualization Results from DBLP.5K:
AutoPhrase results include sets of sorted quality phrases but sometimes they are not "high quality" enough even with high quality scores. We use TF-IDF Score* AutoPhrase quality score to measure the "quality" of a phrase. The TF-IDF and the AutoPhrase separate results will be presented as well. Also the segmentation file highlights the key phrases produced by autophrase to emphasize the phrase location in the text.

	Score	Phrase
0	0.937225	information retrieval
1	0.936936	knowledge management
2	0.927642	data mining
3	0.920133	database design
4	0.919368	relational database
5	0.915095	programming language
6	0.914053	query language
7	0.907747	object oriented
8	0.903430	concurrency control
9	0.901160	machine learning
10	0.900460	natural language
11	0.899760	logic programming
12	0.890344	programming languages
13	0.881059	database management systems
14	0.881037	transaction processing
15	0.877731	database systems
16	0.875018	artificial intelligence
17	0.870824	data structures
18	0.851314	data warehouse
19	0.848263	relational databases

	Score	Phrase
0	0.688071	language
1	0.687766	programming
2	0.686140	management
3	0.685698	approach
4	0.684700	range
5	0.683298	design
6	0.681277	web
7	0.673820	logic
8	0.671405	oriented
9	0.669299	database
10	0.669266	object
11	0.667475	processing
12	0.664019	information
13	0.661126	analysis
14	0.660450	mining
15	0.659142	based
16	0.657607	relational
17	0.657312	reconstruction
18	0.657083	model
19	0.656598	knowledge

	Score	Phrase
0	0.937225	information retrieval
1	0.936936	knowledge management
2	0.927642	data mining
3	0.920133	database design
4	0.919368	relational database
5	0.915095	programming language
6	0.914053	query language
7	0.907747	object oriented
8	0.903430	concurrency control
9	0.901160	machine learning
10	0.900460	natural language
11	0.899760	logic programming
12	0.890344	programming languages
13	0.881059	database management systems
14	0.881037	transaction processing
15	0.877731	database systems
16	0.875018	artificial intelligence
17	0.870824	data structures
18	0.851314	data warehouse
19	0.848263	relational databases

Figure 5: Screenshot of the text presentations based on the data set DBLP.5K

We did some basic analysis about our text, which provide a brief description of the sentences statistics, the token frequency, words number for each line, and the top 20 frequent words in the tokenization steps. On the graphs, the users can see the trend of tokens frequency from the most to the least, and can get a sense of the rough number of tokens in the text. Moreover, it will show the number of words in each line of the text, by a semi log plot.

```
There are 4007 sentences in this input text file. The mean of the input text word length is around 9 for each sentence with the
standard deviation 13. Number of Rare tokens is 6768 (which defined as the the number of tokens is less than 5).
```

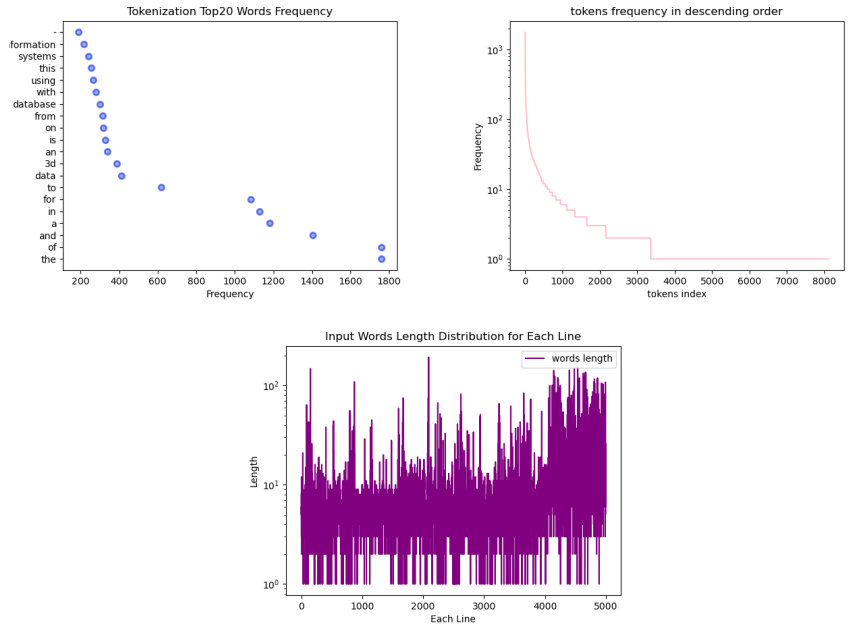

Figure 6: Basic Text Analyses for DBLP.5K.txt

The most important visualization for our web application is the word cloud. The word cloud is plotted with E-chart library based on top 40 word-grams of the AutoPhrase Score* TF-IDF score descending rank. After normalizing, we multiply the value with the AutoPhrase result and select the top 40 high quality phrases, which make the word cloud much more meaningful than only using the AutoPhrase itself only. The phrases shown are much more informative than the AutoPhrase generated phrases, because it not only contains the important phrases, but also shows the frequent phrases from the text input.

Figure 7: Word-cloud based on DBLP.5K.txt

In this quality score distribution, the users can detect the single-word and multi-words phrases score as well. As we can see, the whole performance of multi-words performs worse than a single word, because there are different combinations that can form from uni-gram and the single word is only itself without any adds-on. There might be some special cases when there are some combinations of high-quality phrases that are higher than a single word, which is much more meaningful than the single phrase. It can be explained by the meaning of a multi-word phrase that is much more meaningful than a single word and the multi-word phrase fulfills the concordance criteria than a single word, because a single word does not have any context related word connecting with it.

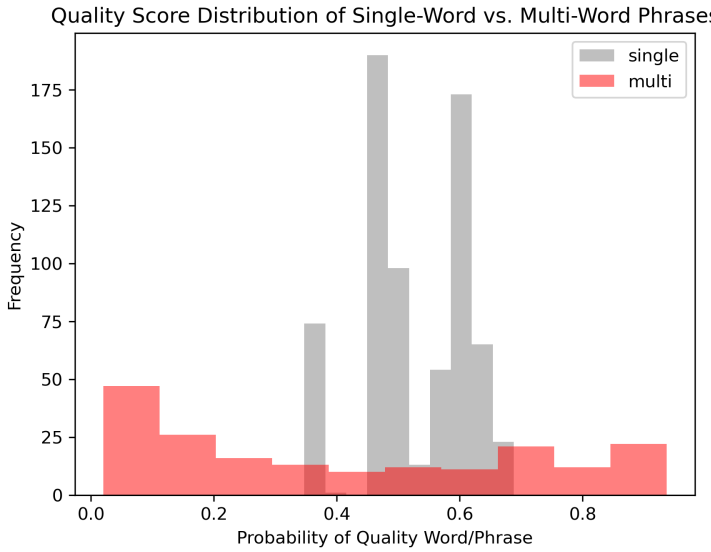

Figure 8: Quality score comparison between multi- and single-word phrases

We also perform sentiment analysis on the input corpus. Figures shown below are the sentiment analysis result for DBLP.5K.txt. Since this is a scientific journal, not surprisingly the polarity and subjectivity is zero for most sentences, meaning most of them are quite neutral and objective. This meets our expectations.

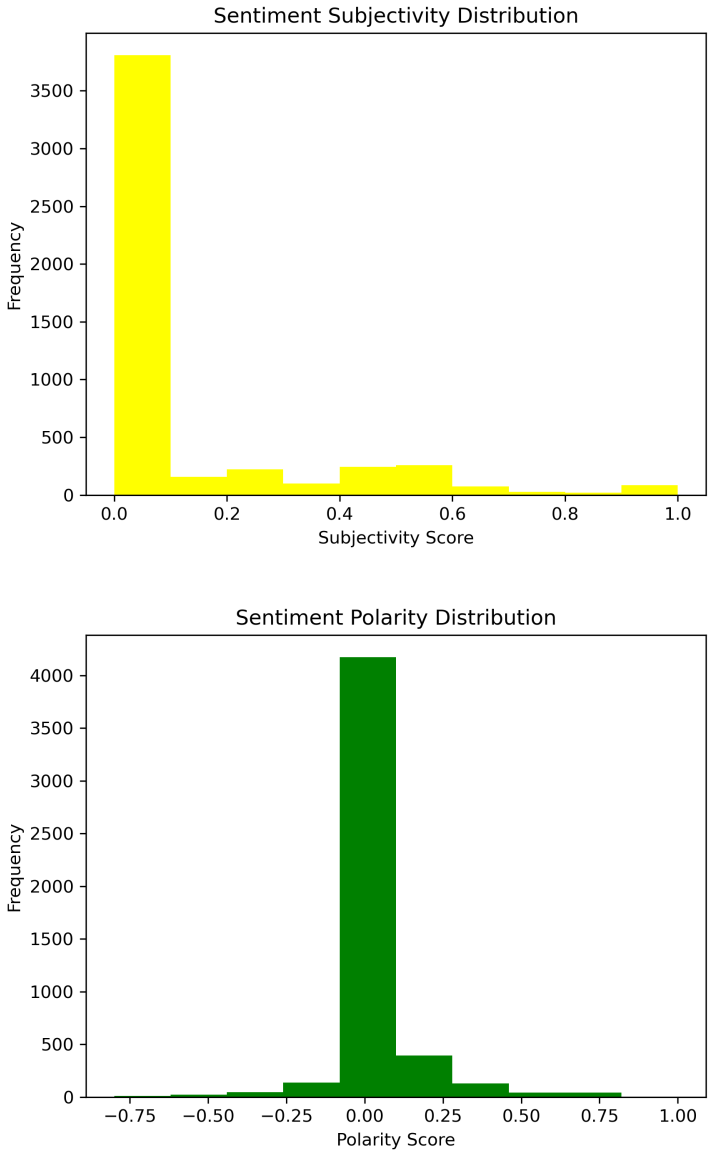

Figure 9: Sentiment Analysis

6 Discussion

We conducted a basic data analysis of the results produced by AutoPhrase, however, even a small sample is quite accurate in producing high quality phrases that are semantically sound and logical. Likewise, as seen through the phrase similarity model, we achieved good similarity scores for phrases that are actually related. This directly shows that the AutoPhrase model can be used in real-world applications – the model can be quite effective in domains such as fraud detection, market research and public sentiment, all of which have a core requirement of effective text mining techniques. Furthermore, I hope our web can help the users to learn the Algorithms of AutoPhrase and get their results much more meaningful and understandable.

7 Future Work

In the future, we plan to optimize the website to:

- reduce run- and load-time
- produce results for significantly large files
- add more data analyses and visualizations
- let user know time taken by AutoPhrase for each model run on the text corpus

8 References

1. Figure 1 borrowed from Automated Phrase Mining from Massive Text Corpora Page3
2. Figure 2 borrowed from Website What Is TF-IDF?
3. Figure 3-9 from the web visualization
4. Jialu Liu, Jingbo Shang, Chi Wang, Jiawei Han, Xiang Ren. *Mining Quality Phrases from Massive Text Corpora.*
5. Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, and Jiawei Han, Fellow, IEEE. *Automated Phrase Mining from Massive Text Corpora.*
6. What Is TF-IDF? *MonkeyLearn Blog*, 10 May 2019, *monkeylearn.com/blog/what-is-tf-idf/: :text=TF%2DIDF%20is%20a%20 statistical,across%20a%20set%20of%20documents*