Yicen Ma
Tiange Wan
Anant Gandhi

AutoPhrase Application Web Visualization Report

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 dataset as an example to the users.

Introduction

Phrase mining - the extraction of high-quality phrases from a large input corpus - is an important and interesting skill to analyze text datasets with. It identifies the phrases instead of an unigram word, which provides a much more understanding of the text. It has various applications such as information extraction, topic modeling, etc. 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 unigram “support”, “vector” and “machine” and it is much more meaningful than any of those unigrams with high frequency of the phrase appearing in the writing (Jialu 1). 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. 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 tables and charts to help them understand the text visually.

Our projects would mainly include:

- A web application, which required the users upload their own input file in the txt format
- An exploratory data analysis, sentiment and TF-IDF analysis, tables/charts visualization for the given input corpus, to help the user understand information from the text in the notebook
- Key performance indicators for the model on the website
- A visualization dashboard for relevant outputs on the website
- Available to change the language base on the website

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. 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.

Methods

AutoPhrase 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.

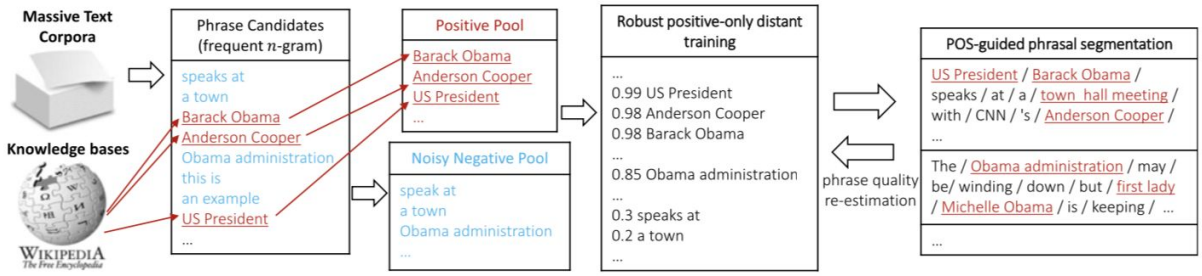

Figure 1. Outline of Automated Phrase Mining using Robust positive only distant training and POS-guide phrasal segmentation

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(ex. default file is from wikipedia) 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.

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 Java. 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.

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 follows:

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

where

$$tf\ (t, d) = log\ (1 + freq\ (t, d)) \qquad idf\ (t, D) = log\ \left(\frac{N}{count\ (d \in D : t \in d)} \right)$$

Figure 2. Algorithm of TF-IDF

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. After normalizing, we multiply the value with the Autophrase result and select the top 20 high quality phrases, which make the word cloud much more meaningful than only using the Autophrase itself only.

Sentiment Analysis

It is one of the uses of Natural Language Processing to study on the subject information and affective states from text. **Sentiment analysis** (or **opinion mining**) is a natural language processing technique used to determine whether data is positive, negative or neutral. We use a Python library called textblob 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.

Result

On the first front web, 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 buttons. After uploading, the text presentation will automatically be updated as presentations. Later, they can click the button "Run AutoPhrase" to start the analysis process. In this demo, we used one dataset called "DBLP.5k", which is the first 5000 lines on the "DBLP" file and it contains the topical keyphrases from the scientific paper domain. We use this dataset, because it can generate results faster than the whole DBLP dataset.

IMAGE

Figure 3. Screenshot of the first front web

On the second front web, 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.

IMAGE

Figure 4. Screenshot of the second front web

On the third front web, it presents the text and visualization results corresponding to their visualization.

IMAGE

Figure 5. Screenshot of the third front web based on the dataset DBLP.5K

Text Analysis and 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.

+ Segmentation txt explanation

IMAGE

Figure 6. Screenshot of the text presentations based on the dataset DBLP.5K

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. 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.

Word Cloud IMAGE

Figure 7. Screenshot of the word-cloud presentation based on the dataset DBLP.5K

+ Add graph explanations

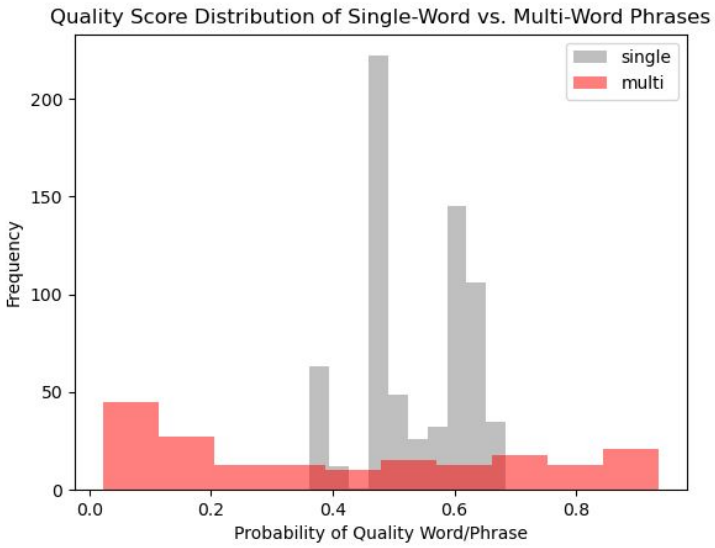

Figure 8. Screenshot of the autophrase presentation based on the dataset DBLP.5K

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.

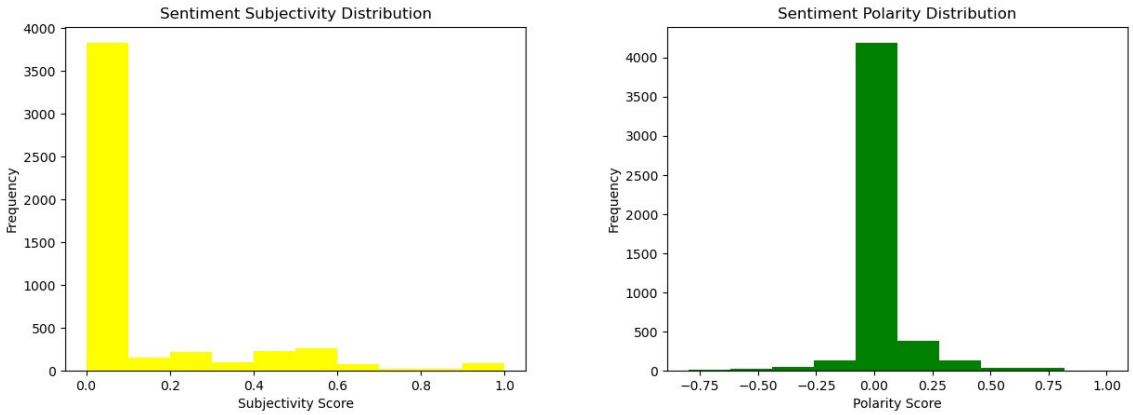

Figure 9. Sentiment Analysis.

Future works

Reference

Figure 1 borrowed from *Automated Phrase Mining from Massive Text Corpora* Page3

Figure 2 borrowed from Website *What Is TF-IDF?*

Figure 3-9 from the web visualization

Jialu Liu, Jingbo Shang, Chi Wang, Jiawei Han, Xiang Ren. *Mining Quality Phrases from Massive Text Corpora*

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, and Jiawei Han, Fellow, IEEE. *Automated Phrase Mining from Massive Text Corpora*

"What Is TF-IDF?" MonkeyLearn Blog, 10 May 2019, monkeylearn.com/blog/what-is-tf-idf/#:~:text=TF%2DIDF%20is%20a%20statistical,across%20a%20set%20of%20documents.

Appendix

DSC180 Capstone Project Proposal- AutoPhrase Website

Yicen Ma, Tiange Wan, Anant Gandhi

Phrase mining - the extraction of high-quality phrases from a large input corpus - is an important and interesting skill to analyze text datasets with. It 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.

This quarter, we were introduced to AutoPhrase, which is a scalable and efficient phrase mining framework. AutoPhrase requires minimal human effort for phrase labeling and can support any language, as long as a solid knowledge base like Wikipedia is known. It takes a large corpus as input and outputs lists of single- and uni-word phrases ranked by their phrase quality. AutoPhrase is useful, interesting and well-structured for real-world applications. However, the framework is not friendly enough for non-technical users or the common people. For those users, running AutoPhrase would be a cumbersome process since it requires:

- Technical knowledge or expertise to run, since it involves the use of GitHub and terminal scripts
- Manual analysis of input corpus and the generated results by the user
- Manual visualization of relevant outputs, in case required

This quarter, we replicated the autophrase structure and process on our GitHub repository. For the second quarter, in order to resolve above issues and simplify the use of AutoPhrase, we propose the creation of a full-stack website as an extension of the AutoPhrase algorithm and analyze the text on the notebooks. The purpose of this project is to help non-tech users understand their text efficiently by utilizing AuoPhrase algorithms and other text analysis techniques.

Our projects would mainly include:

- An exploratory data analysis, sentiment and TF-IDF analysis, tables/charts visualization for the given input corpus, to help the user understand information from the text in the notebook
- Key performance indicators for the model on the website
- A visualization dashboard for relevant outputs on the website
- Available to change the language base on the website

The tools used to approach this application would be Python, HTML, JavaScript, and CSS. We plan to connect the website with the original AutoPhrase repository and with user consent, collect input text data to get an aggregate summary of all input corpora. We will provide the option for the users to provide their own knowledge base if they prefer not to use our default base. We might need to consider possible approaches on dealing with large amounts of input data to train the segmentation model. This would act as motivation to further improve the original AutoPhrase model.For the notebooks, we decided to use TF-IDF, sentiment analysis through charts, word cloud visualization.

Yicen and Tiange will mainly be responsible for the programming and technical issues and report writing while Anant will focus more on web design and deployment. Besides, to make sure we can achieve the proposed goals, we have decided on a tentative schedule, as shown below.

Week	**Task**
Week 1	Implement website skeleton - I
Week 2	Implement website skeleton - II
Week 3	Trying to linking with the AutoPhrase Repo
Week 4	Build the back end
Week 5	Debugging & building the visualization
Week 6	Deployment Connect front and backend of the website, and text analysis

Week 7	Visualizations and text analysis
Week 8	Finalize Project
Week 9	Finalize report

At the end of the quarter, we might plan to launch the website and make it public. By simply entering/attaching their desired input corpus, the user would get the mined phrases instantly, along with a more comprehensive view of the output with our auto-generated dashboard.