Project Report

On

Text Summarization and Document Similarity

By

ANUPREET SINGH	**2017A7PS0955G**
RITURAJ ROY	**2017A7PS0957G**

At

IDS INFOTECH, IT PARK, CHANDIGARH

A Practice School-I Station Of

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

(July,2019)

Acknowledgement

We would like to extend our sincere gratitude towards Mr Pratap K Aggarwal, Managing Director of IDS Infotech Pvt Ltd for giving us the opportunity of interning at his prestigious company as a part of the Birla Institute of Science and Technology, Pilani, PS program.

We are extremely grateful for the support of our Project Mentor Anirudh Munj, without whose guidance and support we could not have achieved the progress we have in data analysis on titanic disaster data set project. He familiarized us with python libraries and methods of data analysis.

We wish to acknowledge the help from our PS Faculty Incharge, Dr. Rajeev Taliyan for guiding us through the Practice School Course.

Birla Institute of Technology and Science

Pilani (Rajasthan)

Practice School Division

Station: IDS Infotech Ltd. Centre: Chandigarh

Duration: From: 21 May 2019 To: 13 July 2019

Date of Submission: 12 July 2019

Title of Project: Text Summarization and Document Similarity

ID Number	Name	Discipline
2017A7PS0955G	Anupreet Singh	CSE
2017A7PS0957G	Rituraj Roy	CSE

Name of Expert: Mr. Anirudh Munj Designation: Project Mentor

Name of the PS Faculty: Dr. Rajeev Taliyan

Key Words: NLP, TF-IDF, Cosine Similarity, Document Classification, Summary Generation

Project Area: Machine Learning, NLP

Abstract: Document Similarity can be calculated using several techniques such Jaccard Similarity, Cosine Similarity, Cosine Similarity with TF-IDF etc. But due to the way cosine similarity works, larger the document is, larger are the chances of errors. So, we take an unconventional approach towards finding similar documents by using

Cosine Similarity on the summary of the documents generated using TF-IDF. This approach will have lesser chances of error as it only considers the broader and more important details of the documents. To reduce the summary size, we score all the sentences accordingly by their importance and use the average score to generate a threshold for the summary. After generating the summary, we use cosine similarity on these to get similarity values. The final output will be a list of files in sorted order based on amount of similarity between these and the input file.

CONTENTS

INTRODUCTION TO NLP

Natural Language Processing (NLP) is concerned with the processing and understanding of the human language. It plays a vital role in text to speech devices, text parsing, document classification, information retrieval and extraction.

Nowadays, it's also used for making powerful search engines, filter spam and obtain semantics of a text in a fast and reliable manner. It's a major component of all voice recognition services like Alexa, Siri, Google and Cortana.

Computers can easily understand the structured form of data like spreadsheets and databases, but human language, texts and speech form an unstructured group of data that is difficult to process. Thus, arises the need of NLP.

There are various stages involved with training a model. These include: -

- Sentence Segmentation: Breaking the pieces of sentence into various pieces.
- Word Tokenization: Breaking the sentence into individual words named as tokens.
- Predicting Parts of speech for each Token: Predicting whether the word is a noun, verb, adjective, pronoun, adverb etc.
- Lemmatization: Extracting the root word from the token.
- Identifying StopWords: Identifying words like 'a', 'an', etc. that provide no descriptive meaning to the document.

- Dependency Parsing: Finding out relationships between the words and how they are related to each other.
- Finding Noun Phrases: Grouping words that represent the same idea. Example the tokens ‘second’, ‘largest’, ‘town’ can be grouped together as single phrase.
- Named Entity Recognition (NER): NER maps the words with real world entities. The entities that exist in the physical world.
- Coreference Resolution: Finding tokens that refer to the same entity. For example, ‘it’ in a sentence may refer to some place in the previous one. These relationships are important for finding the semantics of the document.

After all this, we can create machine learning models that will work according to our requirements. In this Project, we try to make a model that will tell us about the level of similarity between the different documents and also generate summaries for them.

LIBRARIES USED

SCIKIT-LEARN: Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. This library is more focussed on modelling data.

NLTK: The Natural Language Toolkit (NLTK) is used for building python programs that work with human language data for application in statistical Natural Language Processing (NLP). It contains Text Processing libraries for Tokenization, parsing & classification, stemming and semantic reasoning.

MATPLOTLIB: Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc.

PANDAS: Pandas is a popular Python package for data science, and with good reason: it offers powerful, expressive and flexible data structures that make data manipulation and analysis easy, among many other things. The DataFrame is one of these structures.

SEABORN: Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures. It offers API for examining relationships between multiple variables.

NUMPY: NumPy is the fundamental package for scientific computing with Python. It contains powerful tools for

working with N-Dimensional Vectors. It also contains useful algebraic and random number capabilities.

RE: A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. Regular expressions are widely used in UNIX world. The module re provides full support for Perl-like regular expressions in Python.

GLOB: Glob is a general term used to define techniques to match specified patterns according to rules regulated by UNIX shell. WE use glob for file i/o.

GLoVe Vectors: GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

MATHS BEHIND THE PROJECT

TF-IDF

TF-IDF stands for Term Frequency-Inverse Document Frequency, and the TF-IDF weight is often used in data analytics and text mining. The TF-IDF weight for a term is generally calculated by multiplying the TF and IDF of that term. Variations in this calculation of weight can depend on the type of model being used. Individually the terms are calculated as follows:

Term Frequency (TF): **Term Frequency**, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones.

Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document}$$

Inverse Document Frequency (IDF): **Inverse Document Frequency**, which measures how important a term is. While computing TF, all terms are considered equally important. However, it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have

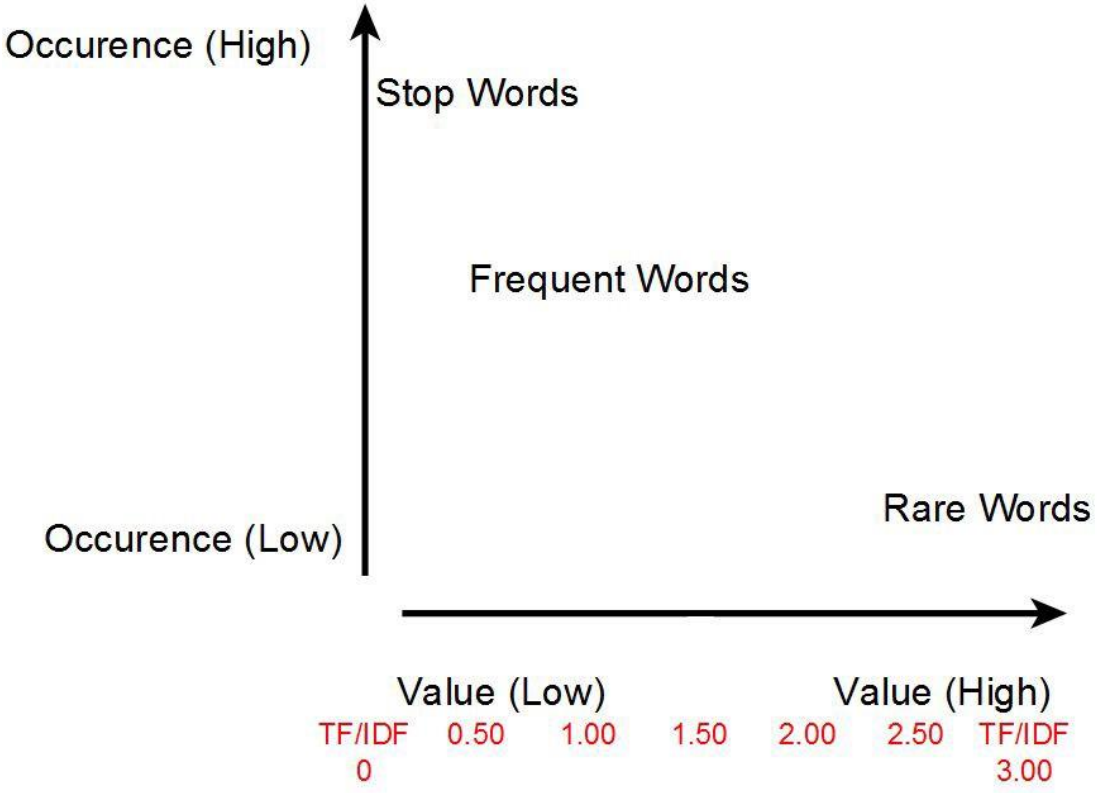
little importance. Thus, we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

$$IDF(t) = Log(\frac{Total\ Number\ of\ Documents}{Number\ of\ Documents\ with\ term\ T\ in\ it})$$

Finally, the TF-IDF of a term is calculated as follows:

$$TF - IDF(t) = TF(t) * IDF(t)$$

The higher the TF-IDF value, the rarer the term is and smaller the value, the more common the term is.



Benefits of TF-IDF include:

1. Removing stop words.
2. Find words with higher search volumes

3. Make sure the terms used are unique and relevant to the user.

COSINE SIMILARITY

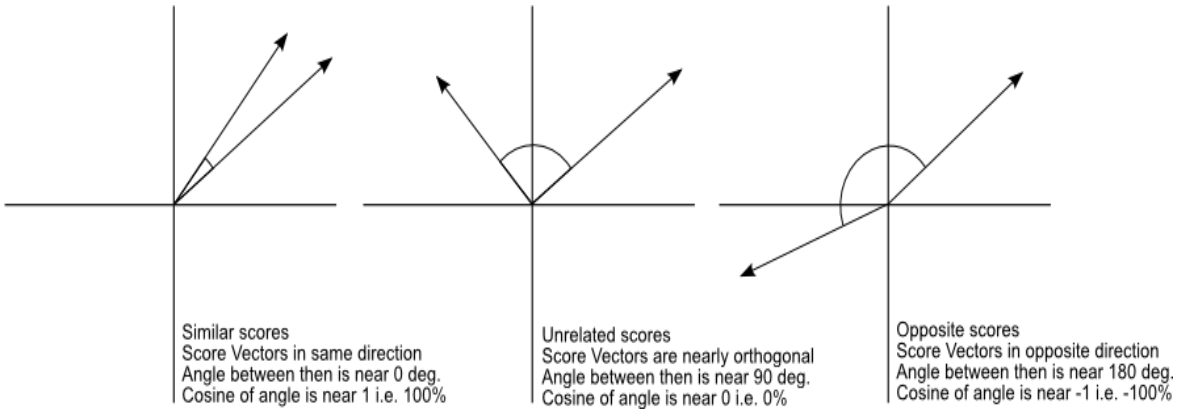
Cosine Similarity is a metric used to find how similar or closely related two documents are. Mathematically, it is the cosine of the angle between two vectors in Three-Dimensional Space. The lower the angle, more the cosine similarity.

Cosine Similarity is extremely advantageous as even if the Euclidian Distance between two documents is large i.e. the size of the documents differs by large amounts, Cosine similarity will give precise and accurate results.

Mathematically,

$$sim(x, y) = (x * y)/(|x| * |y|)$$

where x, y are two vectors in Euclidian Form.



DOCUMENT REPRESENTATION

Several ways exist to model a text document. The most common one being the “Bag of Words” implementation. The Bag of Words model is widely used in text mining and information retrieval. In this implementation, the frequency of each word is its weight, meaning that the terms that appear more, will have a higher importance in describing the document.

Although more frequent words are more important in Bag of Words implementation, this is not necessarily true in practical purposes. For example, words like ‘a’, ‘the’, ’in’ are neither descriptive or important for the Document’s subject. Thus, a more complicated strategy of tokenizing the words is used before applying any algorithms.

Terms are basically words in context of Natural Language Processing. Before vectorizing a term, we apply several standard transformations. First, we remove the stop words. These are words that are non-descriptive for the topic of the document, such as ‘a’, ‘and’, ‘are’ and ‘do’. Several Python libraries provide us with several stop word removal techniques. Second, we use stemming such that words with different ending will map to the same root word. For example, ‘production’, ‘produce’, ‘produces’, ‘product’ will be mapped to the term ‘produc’ after stemming. Finally, we break each text document sentence to produce its summary.

SUMMARY GENERATION

There are two types of summary generation algorithms namely extractive and abstractive. While extractive algorithms work on obtaining the most important and content defining sentences from the document, abstractive summarization is based more on extracting important words and forming sentences on its own. To maintain document consistency, we will be doing extractive summarization.

Firstly, we will be finding the TF (Term Frequency) of each word in the preprocessed document using the formulas mentioned above. Same way, we will be finding the IDF (Inverse Document Frequency) using the same method as mentioned above. Multiplying these two would give us the TF-IDF values for each word. This will be the weight of each word in the document. More common the word is, lower the TF-IDF. Now, we will assign a score to each sentence based on the summation of TF-IDF values of each word in that sentence. This will play a major role in text summarization. Doing this for each sentence, we will get an average sentence score for the document and that score will be used to calculate the threshold to be considered for generating text summary.

The sentences having a score greater than the threshold score will be included in the summary. The output summary is comprised of meaningful sentences instead of random phrases. Thus, this process can be used for summary generation as well. The length of the summary can be modified by multiplying the generated threshold value by any integer ‘x’ that will be user generated. In our case, x=1.

DOCUMENT SIMILARITY

Now that we have generated the summary of all the documents in our dataset and the input document, we will use cosine similarity to find documents that are similar to the input.

Firstly, summary of all the files in the dataset and the input file will be generated using TF-IDF and stored. This will reduce the length of the text files for better cosine similarity functioning.

Now each summary is preprocessed again by removal of stop words and vectorized using GLoVE Vectors. GLoVE Vectors are a set of pre vectorized words that include both nouns and verbs. Next, cosine similarity will be calculated by using the formulae provided above for each document with respect to the input document. The values will be sorted and a list consisting of names of the documents will be returned to the user. This list will be sorted with the most similar document being on top.

A pictorial representation for these is also generated.

Outputs for multiple test cases is shown below.

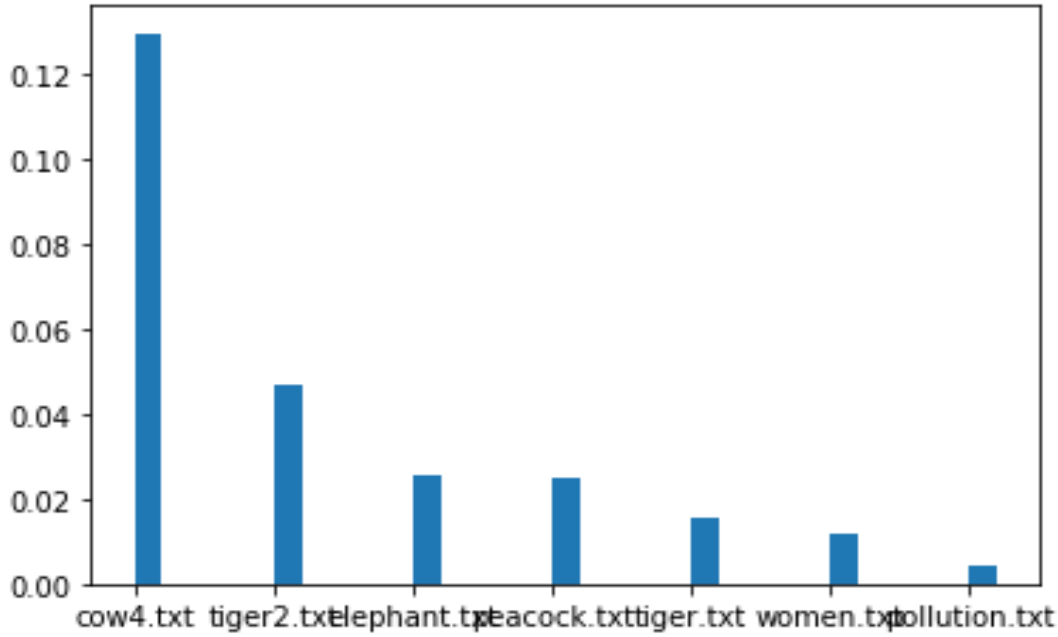
TESTING

Test Case 1: Input File = cow1.txt (an essay on cow)

Folder Name: Docs1

Files in Folder: [cow4.txt, elephant.txt, peacock.txt, pollution.txt, tiger.txt, tiger2.txt, women.txt]

Model correctly predicted that cow4.txt was most similar to cow1.txt. Bar graph is plotted below.

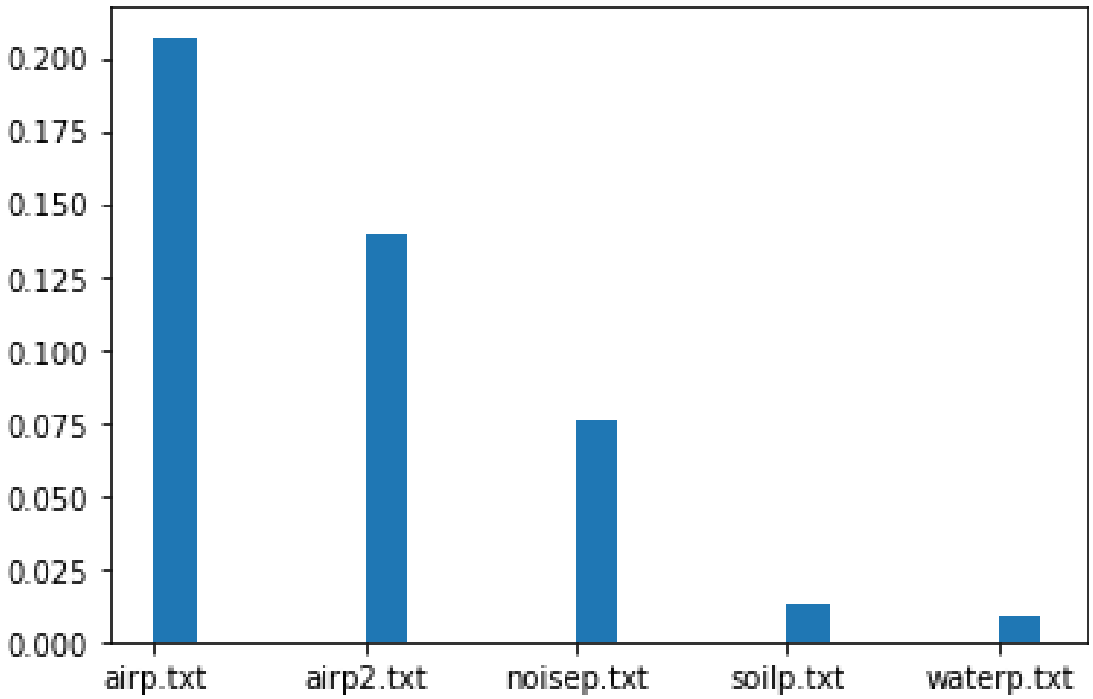


Test Case 2: Input file: airpmain.txt (Article on air pollution)

Folder Name: Docs2

Files in Folder: [airp.txt, airp2.txt, noisep.txt, soilp.txt, waterp.txt]

Model correctly predicted that airpmain.txt was most similar to airp.txt and airp2.txt. Bar graph is plotted below.



Test Case 3: Input file: A file named news.txt which contains a news article refering to the effect of loss suffered by India in the semi finals of the World Cup on the players.

The following summary was generated:

“The death of a dream arrived in the most workmanlike city. Fireworks for some, heartburn for others. It must be hard for him to repeat the "better team won" cliche. Not India. How then will he explain the loss to himself? "It is difficult to explain. Here's an apocryphal tale on Rohit, narrated to yours truly by a senior colleague. Rohit asked. "Honestly, it looks really tough," the scribe replied. I work hard to look effortless, he'd say, but few listened. Then 2013 happened. They are the cricketing equivalence of Sufism. He buried his face twice in his arm before cameras shifted their focus. There could be a tear or two too, but no amount of high-resolution lenses will tell you how broken he must be. He has been infuriating, excruciating, endearing, but in pain? Sport does that to you.”

The original article consisted of 7649 characters whereas the summary consists of 798 characters.

Test Case 4: A file named causesofclimatechange.txt which contains an article on the causes of climate change.

The following summary was generated:-

“Causes of climate changeWhat is the most important cause of climate change?Human activity is the main cause of climate change. People burn fossil fuels and convert land from forests to agriculture. Since the beginning of the Industrial Revolution, people have burned more and more fossil fuels and changed vast areas of land from forests to farmland.Burning fossil fuels produces carbon dioxide, a greenhouse gas. It is called a greenhouse gas because it produces a “greenhouse effect”. Other greenhouse gases, such as nitrous oxide, stay in the atmosphere for a long time. Other substances only produce short-term effects.Not all substances produce warming. They do this by affecting the flow of energy coming into and leaving the earth’s climate system.Small changes in the sun’s energy that reaches the earth can cause some climate change. Emissions of other substances that warm the climate must also be substantially reduced. This indicates how difficult the challenge is.What is climate change?Climate change is a long-term shift in weather conditions identified by changes in temperature, precipitation, winds, and other indicators. However, its long-term state and average temperature are regulated by the balance between incoming and outgoing energy, which determines the Earth's energy balance. Any factor that causes a sustained change to the amount of incoming energy or the amount of outgoing energy can lead to climate change. Different factors operate on different time scales, and not all of those factors that have been responsible for changes in earth's climate in the distant past are relevant to contemporary climate change. Factors that cause climate change can be divided into two categories - those related to natural processes and those related to human activity. Of these, the two factors relevant on timescales of contemporary climate change are changes in volcanic activity and changes in solar radiation. In terms of the Earth's energy balance, these factors primarily influence the amount of incoming energy. Volcanic eruptions are episodic and have relatively short-term effects on climate. Since the beginning of the Industrial Revolution, these human influences on the climate system have increased substantially. These in turn can influence both the amount of incoming energy and the amount of outgoing energy and can have both warming and cooling effects on the climate. Other substances have shorter atmospheric lifetimes because they are removed fairly quickly from the atmosphere. Therefore, their effect on the climate system is similarly short-lived. Together, these short-lived climate forcers are responsible for a significant amount of current climate forcing from anthropogenic substances. However, reducing emissions will quite quickly lead to reduced atmospheric levels of such substances. That is, the warming we have experienced to date would have been even larger had it not been for elevated levels of sulphate aerosols in the atmosphere.”

The original article consisted of approximately 9000 characters whereas its summary consists of 3000 characters.

On comparingthe summary with the original article it was found that the model gave higher proirities to the main sub headings and included them in the summary without fail.

FUTURE SCOPE

The project can be used for summarization of all types of text documents and can also play the role of searching for text documents with similar data.

Document clustering can be carried out by clubbing together files with similar data depending on the pairwise cosine similarity values between the text documents.

This can also check for plagiarism as a cosine similarity value close to 1 shows a high probability of the two documents to be copies of each other,using this property it can also act as a document searching model.

This project can also be used for searching similar types of documents when provided with a large data set.This application can also be utilized to find out whether a file is already present in a folder of a large number of files under a different name.Therefore allowing us to prevent repetition of data in a single directory.

BIBLIOGRAPHY

1. NumPy: https://www.numpy.org/devdocs/
2. Pandas: http://pandas.pydata.org/pandas-docs/stable/
3. Jupyter Notebook
4. Seaborn: https://seaborn.pydata.org/
5. Matplotlib: https://matplotlib.org/users/index.html
6. Scikit-learn:https://scikit-learn.org/stable/documentation.html
7. NLTK: https://www.nltk.org/
8. GLoVe Vectors: https://nlp.stanford.edu/projects/glove/
9. RE: https://docs.python.org/3/library/re.html
10. Glob: https://docs.python.org/3/library/glob.html