Text Summarization using LSA with truncated SVD

CS8011
MACHINE LEARNING PROJECT

Under the guidance of:

Dr. Kusum Kumari Bharti

Team Members:

Ashesh Dubey - 2019033
Garvit Gupta - 2019060

Abstract

Document summarising is a task in natural language processing that works with extensive textual material to provide short and fluent summaries that include all of the document's significant content. The branch of NLP that deals with it is automatic text summarizer, which is the task of turning long textual content into short fluent summaries. There are two common methods for summarising text with an automatic text summarizer: one is to use an extractive text summarizer, and the other is to use an abstractive text summarizer. Topic modelling is an NLP task that extracts the relevant topic from the textual content. Latent Semantic Analysis (LSA) with truncated SVD is one such method that extracts all relevant topics from the text. After prediction, we then compare the resultant summaries against different evaluation metrics and deduce the most suitable one.

Introduction

In recent times, the global internet population has grown at a phenomenal rate, with over 4.6 billion active users. The increase in the number of people using the internet in recent decades has increased data, primarily text-based data. Some of the biggest platforms that are dealing with textual data overhead are social networking, news publishing platforms, and search engines. Furthermore, these data are unstructured and contain millions of pieces of significant information that must be managed and handled correctly. Machine Learning with Natural Language Processing aids in the labelling and organisation of extremely unstructured textual material.

As the amount of text data generated every day grows, NLP has become more vital in making sense of it. Any organisation must be able to make data-driven decisions because data is generated and saved with every internet click. Automatic text summarizer and Topic Modelling are two fields of Natural language processing that are commonly used to reduce massive amounts of data in data centres. This reduces the quantity of data while maintaining vital information and aids in data organisation. An automatic text summarizer is a sub-domain of Natural Language Processing that is commonly used to condense voluminous textual material into a more manageable format. Because automatic text summarizers reduce data size, they also save reading time and enhance the amount of useful information in a given region.

In this project, we have demonstrated the experiment in which we are going to make the article summaries and news headlines using the Extractive Text Summarizer approach by using the LSA algorithm truncated using SVD. The “News Summary” dataset is used which contains Headline, Summary, Article text, Date and the Link to that news article. Final scoring is made by comparing the positional embeddings using Cosine Similarity of each sentence to that of the topics extracted using LSA along with other evaluators like ROUGE-L and Singular Value Decomposition. Therefore in this way semantics of each sentence can

be captured more accurately. For the scope of this project, we will be strictly sticking to only Machine Learning algorithms and will not be considering any Deep Learning approaches. All this is discussed in detail in the next sections.

Background and problem description

Nowadays there is a plethora of information available on a variety of topics on the internet. Due to excessive contribution by the people, this information does not necessarily serve the needs of a user. News articles are a major part of this information on the Internet. In today's busy life, people do not spend a lot of their time reading news articles. Often these articles tend to be click-bait and unnecessarily lengthy which ultimately wastes users' time. Automating the filtering of misinformation from information is an extremely difficult task. So, our approach to overcoming this problem is by reducing the time users waste knowing about a subject, by reducing the content on which the user spent their time on. This reduction of content is achieved with the help of text summarization.

There are mainly two approaches to automatic text summarization 1) Extractive text summarizer and 2) Abstractive text summarizer. Extractive text summarizer involves extracting relevant phrases, keywords and sentences from the source textual document without making changes to them, further it combines them to make a summary. On the other hand, an abstractive text summarizer is an approach in which the model is trained using deep learning algorithms which help it create new phrases and keywords that relay the information of the original text, and then it combines with some phrases or sentences from original documents to make up the summary. This project has been based on an extractive text summarizer.

There are many ways in which extractive text summarization can be performed in NLP, like using the text rank method in which text summarization is done by capturing relevant sentences from the long text document by sentence embedding and scoring them using cosine similarity and then combining top sentences to form summary, another method is using topic modelling which captures the relevant sentences based on the topic which long textual document relay on. There are multiple methods for extractive text summarization in NLP, including the text rank method, which captures relevant sentences from a long text document by sentence embedding and scoring them using cosine similarity and then combining top sentences to form a summary. Another approach is the topic modelling method, which captures relevant sentences based on the topic that a long textual document relays on.

In NLP, topic modelling breaks down a text corpus to uncover semantic structure within the text and then extracts the document's subjects. Various topic modelling techniques exist, including LSA with shortened SVD, LDA, and others. We will use the previous method LSA (Latent semantic analysis) with reduced SVD in this project. LSA is a model for extracting and representing the context meaning of words to compute the similarity between words, phrases, or the entire document. LSA analyses a text document using TF-IDF and then learns themes by performing matrix decomposition on the document term matrix using singular value decomposition. LSA is used for reducing the dimension of the matrix to extract

the topic from the text document more precisely. Truncating it with SVD will make sure to only keep the top and most relevant information from the sentences.

Related Work

Automatic text summarizer is a vast field for research in Natural language processing. There had been much previous research work that had contributed to the development of automatic text summarization using topic modelling. Although Topic modelling came into existence in 2003 but gains a large amount of popularity in today's world, It aims to extract the topics from text documents by understanding statistical relations among topics. These topics can be further used in Automatic text summarizers like the Extractive text summarizer that is discussed in later sections.

Summarization using topic modelling aims to develop text summaries using 2 methods as discussed by Yihong Gong and Xon Liu in which text summaries are generated by ranking the sentences and extracting relevant sentences from text documents and the second method is using LSA(Latent Semantic Analysis) to identify relevant sentences from the text document then combining them to form summaries. Another approach to text summarization using topic modelling is discussed in, in which text summarization is performed using term sentence matrix, in this topic embedded in the sentences of the document helps in sentence selection for automatic Extractive text summarizer approach.

Tong, Zohu &Zhang, Haiyi in 2016 published a paper in which they use the LDA (Latent Dirichlet allocation) topic modelling algorithm for extraction of a set of keywords from a text document, then clustering the set of recurring keywords in the group of sentences for text summarization. Although this model performed well in Wikipedia articles still the major missing point is the semantic structure of topics in each sentence of the document.

Proposed Solution

The main idea behind the project is to generate the extractive text summary using LSA (Latent semantic analysis) topic modelling and SVD (Singular value decomposition) in an efficient way such that the generated summary contains all the relevant information about the text document.

Pre-Processing

Text preprocessing is one of the major tasks of Natural Language processing that performs the task of cleaning and removing the ambiguity from data for the efficient performance of an NLP algorithm. As algorithm learns weights more accurately when given the right data. In our proposed model text preprocessing is performed by splitting the text document into the list of sentences and then passing these list of sentences into the text preprocessing.

All the text in the data is normalized to the same sequence of data, i.e. all data must be in the same manner either all letters in the lower case or all letters in the upper case. For the betterment of the model here all letters are normalized to the lower case. For an algorithm to learn weight properly and for capturing semantic aspects of the text removing punctuation is one of the necessary tasks. Punctuation removal involves the removal of punctuations like '?', '!', '/' etc. Stop words include words like 'a', 'an', 'in', 'is' etc. which are the useless data of any text document that doesn't contribute to the semantic as well as syntactic aspects of the text document. These are removed by first tokenizing the sentence obtain by the punctuation removal then removing stop words by analyzing and removing each word that is present in the predefined list of stop words. Finally, we also Lemmatize the words. It involves morphological analysis of the words and aims to remove the ending of the words that constitute the same meaning. Returning to the root word, for eg: running is changed back to run.

Model Formation

LSA topic modelling using Truncated SVD is used to extract the topic by generating the matrix that represents conceptually related sentences or documents and corresponding words. LSA assumes that these words will occur in similar pieces of text if they occur together and represent the same semantic meaning. SVD reduces this matrix for reducing dimension and noise, thereby capturing only relevant topics from the textual document. In our project, sentences from a long textual document are passed to LSA topic modelling for finding the topic on which the textual document is based on. For document D we name the number of sentences in it as (i) and these sentences are passed through LSA topic modelling. LSA creates a term-document matrix for semantic understanding. As the size of the textual document increases the dimensions of the LSA term matrix proportionally increase, therefore to extract the data containing relevant information or extract the topic from the term matrix SVD is used along with LSA. The process starts with the creation of a term by sentences matrix A = [A1, A2, …, An] with each column vector A_i, representing the weighted term-frequency vector of sentence i in the document under consideration. If there are a total of m terms and n sentences in the document, then we will have an m × n matrix A for the document. Since every word does not normally appear in each sentence, matrix A is sparse. The equation of reducing term-document matrix 'A', containing m sentences in rows and n unique words in columns into k number of topics using SVD is:

$$A = USV^T$$

where $U = [u_{ij}]$ is an m × n column-orthonormal matrix whose columns are called left singular vectors; S = diag(σ1, σ2, …, σn) is an n × n diagonal matrix, whose diagonal elements are non-negative singular values sorted in descending order, and $V = [v_{ij}]$ is an n × n orthonormal matrix, whose columns are called right singular vectors.
In A, the dot product of SV^T will be used for truncation. For each sentence vector in matrix V (its components are multiplied by corresponding singular values) we compute its length. The reason for the multiplication is to favour the index values in the matrix V that corresponds to the highest singular values (the most significant topics). Formally:

$$s_k = \sqrt{\sum_{i=1}^{n} v_{k,i}^2 . \sigma_i^2}$$

where s_k is the length of the vector of k^{th} sentence in the modified latent vector space. It is its salience score for summarization too. n is the number of dimensions of the new space. This value is independent of the number of summary sentences. Finally, we put into a summary the sentences with the highest values in vector s.

Apart from Cosine Similarity and ROUGE, we also propose to evaluate our model over SVD similarity that takes into account the first singular vector of the U matrix. This evaluation method has the following advantage over the previous ones. Suppose, an original document contains two topics with the relatively same significance (corresponding singular values are almost the same). When the second significant topic outweighs the first one is a summary, the main topic of the summary will not be consistent with the main topic of the original. Taking more singular vectors (than just one) into account removes this weakness.

Evaluation Study

We have first used cosine similarity to evaluate the similarity between the predicted summary and actual summary. While cosine similarity measures the similarity between two vectors of an inner product space. It tells us whether the 2 sentences point in the same direction or not. We also decided to evaluate based on the ROUGE score or the Recall-oriented understudy for Gisting Evaluation, which is used for evaluating text summaries. We have used the rouge-l parameter for comparing which finds the longest common subsequence between 2 sentences and calculates precision(P), recall(R) and F-Measure using these formulae.

Precision (P) = $N_{overlapping}$ words / $Total_{summary}$ Words

Recall (R) = $N_{overlapping}$ words / $Total_{refrence}$ Words

F-Measure = (2XPXR) / (P+R)

Where $N_{overlapping}$ words are the total number of the overlapping words and Total words in machine-generated summary. And where $Total_{refrence}$ Words is the total number of words in reference summary or original summary.

But Rouge score is useful in evaluating abstractive based summaries. For evaluating the extractive based summary we will use the SVD similarity score between the predicted summary and the actual summary to calculate how much they resemble the same topic.
For calculating the SVD similarity score we will first factorise the words-topic matrix and then use the U matrix. We will first normalize the first column of U matrices and then find the angle between these 2 vectors.

$$cos\,\phi = \sum_{i=1}^{n} u_1(i) * u_2(i)$$

where u_1 is the first left singular vector of the full-text SVD, u_2 is the first left singular vector of the summary SVD (values, which correspond to particular terms, are sorted up the full-text terms and instead of missing terms are zeroes), n is a number of unique terms in the full text.

Results

	Cosine Similarity Score	SVD Similarity Score	Rouge-l F measure	Rouge-l Precision	Rouge-l Recall
Mean	42%	76%	31%	40%	31%
Median	41	77%	30%	36%	30%

Tabular Representation of Evaluation Metrics over summary: Cosine Similarity, SVD Similarity and ROUGE-L scores of F measure, Precision and Recall

Discussion

The experiment is conducted to figure out the best possible way to solve the extractive text summarization using topic modelling. For the purpose of evaluation, we have selected the news articles for text summarization from the “News Summary” dataset. We then run all the articles through the summarizer and generate corresponding 1 line headlines and summaries. Their cumulative score is calculated using Cosine Similarity, SVD Similarity and the rouge evaluation method where ROUGE stands for the Recall-oriented understudy for Gisting Evaluation. A standard rouge package includes several measures to quantitively compare the machine-generated summary and the original summary. For this project we have only used ROUGE-L as there is no need to define the unigram length, it automatically selects the longest common subsequence. While cosine similarity measures the similarity between two vectors of an inner product space. It tells us whether the 2 sentences point in the same direction or not. But it is not the correct way to find evaluate the summary of an article because it compares on the basis of words. Furthermore, the Rouge score is useful for evaluating abstractive based summaries. Since the Extractive approach doesn't form new sequences or words, this also isn’t a good evaluator for our project. This brings us to the SVD similarity evaluator. Suppose, an original document contains two topics with the relatively same significance (corresponding singular values are almost the same). When the second significant topic outweighs the first one a summary, the main topic of the summary

will not be consistent with the main topic of the original. Taking more singular vectors (than just one) into account removes this weakness as done by the SVD approach. Since SVD uses topic-based factorization it will be a better evaluation criterion to calculate how much the predicted and actual summaries are close to a topic.

Conclusion and Future Direction

This project explains the use of sentences keyword extractor and LSA topic modelling on a text document resulting in extracting useful sentences from a text document that contains a useful amount of information about the topic on which the text document is based. The algorithm was able to capture the semantic meaning of the topic word vectors in contrast with their semantic meaning to extract the sentences containing a relevant number of sentences regarding those topics. LSA topic modelling using Truncated SVD results in generating more accuracy for text summarization than LSA topic modelling alone. This tells us that each sentence contributes more towards capturing semantic aspects of text than the whole text alone.

Further, we conclude that SVD Similarity is a better evaluation metric than Cosine Similarity and ROUGE scores as the semantic aspects of text rely on a combination of topics rather than singular words or exact subsequences. This illustrates that instead of creating clusters from received topics to extract sentences containing semantic and syntactic aspects of the text, we could generate a more accurate summary by comparing the semantic aspect of each sentence to the topics from the LSA model to generate a fluent and coherent summary. The future aim of this project is to generate a summary more accurately; also, using the proposed algorithm in an abstractive text summarizer where the machine generates a summary in its own language using Deep Learning approaches. This may result in greater accuracy which can then be studied again using different evaluation metrics to observe which one comes out as the better approach.

References

1. M. Ramina, N. Darnay, C. Ludbe and A. Dhruv, "Topic level summary generation using BERT induced Abstractive Summarization Model," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 747-752, doi: 10.1109/ICICCS48265.2020.9120997.
2. Steinberger, Josef & Jezek, Karel. (2004). Using Latent Semantic Analysis in Text Summarization and Summary Evaluation.
3. M. Ramina, N. Darnay, C. Ludbe and A. Dhruv, "Topic level summary generation using BERT induced Abstractive Summarization Model," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 747-752, doi: 10.1109/ICICCS48265.2020.9120997.

4. Introduction to Text Summarization with ROUGE Scores (https://towardsdatascience.com/introduction-to-text-summarization-with-rouge-scores-84140c64b471)
5. Scikit-Learn cosine similarity (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html)
6. Understanding LSA vis SVD (https://matpalm.com/lsa_via_svd/intro.html)