Automatic Summarization of Scientific Articles

Contents

Abstract

We propose a pipeline for summarizing scientific articles. The pipeline consists of a baseline algorithm, TextRank, to rank sentences within a document based on relevance of the words within each sentence. To improve the selection of important sentences, each ranked sentence is then given as input to an SVM classifier to decide whether the sentence should be included in the summary or not. The ROUGE results show an increase in the scores over the baseline and we also present an analysis of the why our algorithm performs better than the baseline.

Introduction

A summary is a text that represents and preserves the important informaiton conveyed by a document in a concise manner. Automatic summarization has been an area of research for the last couple of decades and has been tackled with different approaches. In this paper we will address the problem of summarization of scientific documents and put forward a solution in the form of a pipeline to process scientific articles and output their summaries.

In order to summarize a document, it is first necessary to identify what pieces of text within the document are important enough to represent the information conveyed by the document. Such pieces of text can range from single words (motivated by the problem of keyword extraction) to clauses and sentences within the document to be summarized. In this paper, we will focus on complete sentences for the reason that a sentence represents a complete coherent line of thought compared to using clauses. So it would be easier to plug the chosen sentences into the summary without worrying about any significant loss of continuity.

Another point to be considered is that summaries are generaly meant to be concise and limited in the number of words. It follows that not all information, that has been identified as important, might fit into a summary. Hence, there is a need to rank and choose the most important pieces of informaiton (sentneces) to be included into the summary.

Eventually, all the high ranked sentences can be clubbed together and presented as a summary in what is termed as *extractive summarization*. A further improvement over creating extracts for summaries is to create an abstract of the assembled text which falls in the domain of language generation.

In this study, we will focus on creating extractive summaries. We started our study by building a baseline to rank sentences based on the relevance of words that occur in each sentence. The measure of relevance depends intuitively on how often words appear in different sentences in the document under consideration. Based on this overlap of words among different sentences, the ranking algorithm gives a high score to a sentence if it has a large number of overlaps with other sentences in the same document. The details of this algorithm , TextRank, proposed by [**?**], are discussed in Section 3. To improve the information content of the summary,

********Describe issues and challenges********

Related Work

Methodology

Problem Description

Since this project is focussed on extractive summarization, the aim is to select sentences from the source article that could be considered worthy of being included in the summary. So I decided to approach this as a problem of classification of sentences (in the source document) into one of the two classes: 'belonging' to summary and 'not belonging' to summary.

Baseline

The first step was to build a basic system that could be the basline for further experiments and improvements. Another factor for consideration was that the aim of the project was to create an extractive summary consisting of sentences from the same document that needs to be summarized. After studying various approaches used by researchers for single document summarization, I decided to use an unsupervised ranking algorithm. Following is a brief description of the TextRank algorithm, as introduced by [**?**], to explain how a graph based raking algorithm can be applied to rank sentences within a document.

TextRank

The basic idea implemented by a graph-based ranking model is that of 'voting' or 'recommendation'. When one vertex links to another one, it is basically casting a vote for the other vertex. The importance of a vertex is based on the number of votes casted towards that vertex.

Formally, let *G = (V, E)* be a directed graph with the set of vertices *V* and set of edges *E* where is a subset of *V x V*. For a given vertex V_i, let $In(V_i)$ be the set of vertices that point to it, and let $Out(V_i)$ be the set of vertices that V_i points to. The score of vertex V_i is defined as follows (*Brin and Page []*),

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

where *d* is the damping factor that can be set between 0 and 1. The factor *d* is generally set to *0.85* and this the value that was used for my experiments as well. Starting with arbitrary

values assigned to each node in the graph, the computation iterates untill convergence below a given threshold is achieved. The final score associated with each vertex represents the *importance* of that vertex within the graph.

Two modifications were introduced by *Mihalcea []* to apply this algorithm to documents for ranking sentences. The first modification was to define the above mentioned algorithm for undirected graphs in which case, the out-degree of the vertex is equal to the in-degree of the vertex. The other modification was to incorporate the notion of *strength* of the connection between any two vertices as a weight w_ij added to the corresponding edge that connects the two vertices. Consequently, the new formula that takes into account these two modifications and gives the weighted score is

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_j i}{\sum_{V_k \in Out(V_j)} w_j k} WS(V_j)$$

TextRank for Summarization

To apply this algorithm for ranking sentences, a graph a built in which the sentences (from the source document) represent the vertices of the graph, and the edges between these vertices is defined by the sentence 'similarity'. Here, 'similarity' is a measure of the content overlap between the two sentences that connected by that edge. This overlap of content between any two sentences can be defined by the number of common tokens between the lexical representations of the two sentences.

For the purpose of the work described in thesis, this concept of similarity between sentences within a document has been implemented in the following way. A matrix of *tf-isf* (term frequency - inverse sentence frequency) is contructed for the entire document. This tf-isf matrix is then multiplied by its transpose to obtain the similarity matrix with dimensions equal to the number of sentences in the entire document, and each value representing the similarity between the two sentences used to index that value in the matrix.

After running the ranking algorithm over this graph untill convergence, the sentences are sorted in decreasing order based on the final score. The top ranked sentences are selected for inclusion in the summary. The performance of this algorithm on scientific articles as baseline is documented in detail in the Experiments and Results section of this report. This is chosen as the baseline as it has been shown to perform reasonably well on unstructured text, for example news articles.

Performance Evaluation of Baseline

One of the well known summarization systems, MEAD, extracts random sentences from the text that has to be summarized. In ..., this basic scheme has been shown to perform well in case of summarizing news articles. To evaluate the performance of the baseline, I compared it with MEAD random based algorithm and found the above mentioned baseline to be better than MEAD. The results of the comparision have been documented in the Experiments and Results section. This can be intuitively justfied as well. MEAD random algorithm is used to

summarize smaller pieces of text (news articles) where as the aim of this project is to summarize scientific articles that are generally quite long. Considering that the corpus used for this project consists of scientific articles from the ACL Anthology, these articles are generally about a specific study and elaborate on the methodology of the conducted experiments, the results obtained, conlusion and possible discussion arising from conducting the experiments. These also include an "Introduction" section as well as a "Related Work" section to describe the work of others who have addressed the same or similar problem.

Although the performance of the baseline system was better than that of MEAD summarization system, further qualitative analysis of the summaries generated by this baseline showed that these summaries included sentences which varied in degree of suitability to be included in the ideal summary. Here ideal summary would be one that is similar in content to the summary created by humans as in the gold standard that was used in this project (more on the gold standard later). Each of the extracted sentence from this baseline was objectively judged as to whether its contents were important enough for that sentence to be included in the ideal summary. It was found that ...% of sentences from all the extracted sentences were not suitable in this respect.

To do (??)

The evaluation of baseline helped in understanding that it is not enough to rely on an algorithm that ranks sentences based on content overlap with other sentences. The importance of the words within the sentence should also be considered. I decided to explore how the importance of words appearing in a sentence, with respect to the article being summarized, relates to whether that sentence should be included in the summary or not. It is also important to limit the examination to words that contribute to the immediate (central) meaning conveyed by the sentence as opposed to considering adjunct clauses or phrases that do not contributet much to the central meaining of the sentnece. The following section describes how dependecy parse trees can help with such a study.

Dependency Parses

Experiments and Results

Discussion

Conclusion