Project Report on

Document Summarizer

Guided By:- Dr. Vasudeva Varma

Mentored By:-

Litton J Kurisinkel

Submitted By:

Group-27

Ashrith Jalagam(201202126)

Shefali Soni(201405619)

Aditya Lunawat(201405559)

1) Abstract:

Among many approaches to calculate the relevancy of a document with respect to given domain, one is to get its summary and then decide its relevancy. We should also keep in mind that, there are various summarizers present for text summarization of a document. We aim to provide a comparision of the summaries created by some of these available summarizers on basis of relevancy to computer science domain. Document Summarizer also provides a common platform to summarize text documents of various formats, using any one of the set of summarizers.

2) Introduction:

Radev et al. (2002) define a summary as "text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that". Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. The definition captures three important aspects that characterize research on automatic summarization:

1. Summaries may be produced from a single document or multiple documents
2. Summaries should preserve important information
3. Summaries should be short

These summaries created by automatic summarization techniques may lack some important information from original documents or may have noise. Also some of the techniques might work better in certain domain/scheme but are bad in other. Thus an evaluation system is required to grade the performance of results obtained from these automatic summarizers.

Our project aims towards creation of summary of a single document with any specific summarizer and quantifying and reporting the match of the created summary with computer science domain. Prime task of Document Summarizer is to get 2-5 lines of summary(or snippet) of given document or URL and describe its relevance to "Computer Science" domain.The Document Summarizer can create summary of Document of any format eg: Text (.odt/.txt/.docx etc) or a URL , provided by user. The input document is parsed and the Text Content is extracted. 2 to 5 lines of summary is prepared for the extracted document using the summarizer chosen by the user. Then, the summary is given as input to the Document Relevance calculator which finally generates the document relevance

with respect to "Computer Science" domain as output.

The rest of the report is organized as follows: Section 2 describes the approach to solve the problem, Section 3 Problem Statement and scope of project of project and working of the system, Section 4 Approach to solve the problem , Section 5 deals with architecture of system and Section 6 explains working model of system. Section 7 provides experimental results, 8 and 9 sections provide an insight of applications and future work related to system, Section 10 concludes the preject report and Section 11 enlists refrences in the project.

3) Problem Statement and Scope:

Can we quantify the given documents relevancy with respect to given domain? Is there any way to decide which summarizer works better in “Computer Science” domain?

- Several summarizers makes it difficult to judge which summarizer suits the best for a scenario. So get the platform for deciding it.
- Ability of the platform to test different summarizers based on a domain helps the developers to make a choice.
- This can be achieved by rating the documents based on their relevancy factor.

Scope of our project is for any given document (file or URL) containing valid textual content and a list of summarizers, quantify the relevancy of the document wrt “Computer Science” domain for selected summarizer.

4) Approach:

For Summarization of differnt text formats of documents we must first get the texual content from the input documents. Also the input could be a URL, thus we have to extract text from the URL. The extracted text format will provide a common input format for all the summarizers. This text data will be sent to summarizer selected by user and a summary is created.
Next we need get information related to computer Science" domain present in web(wikipedia). This will serve as reference to calculate relevancy of document. To do this we need to create a data model for "Computer Science" domain

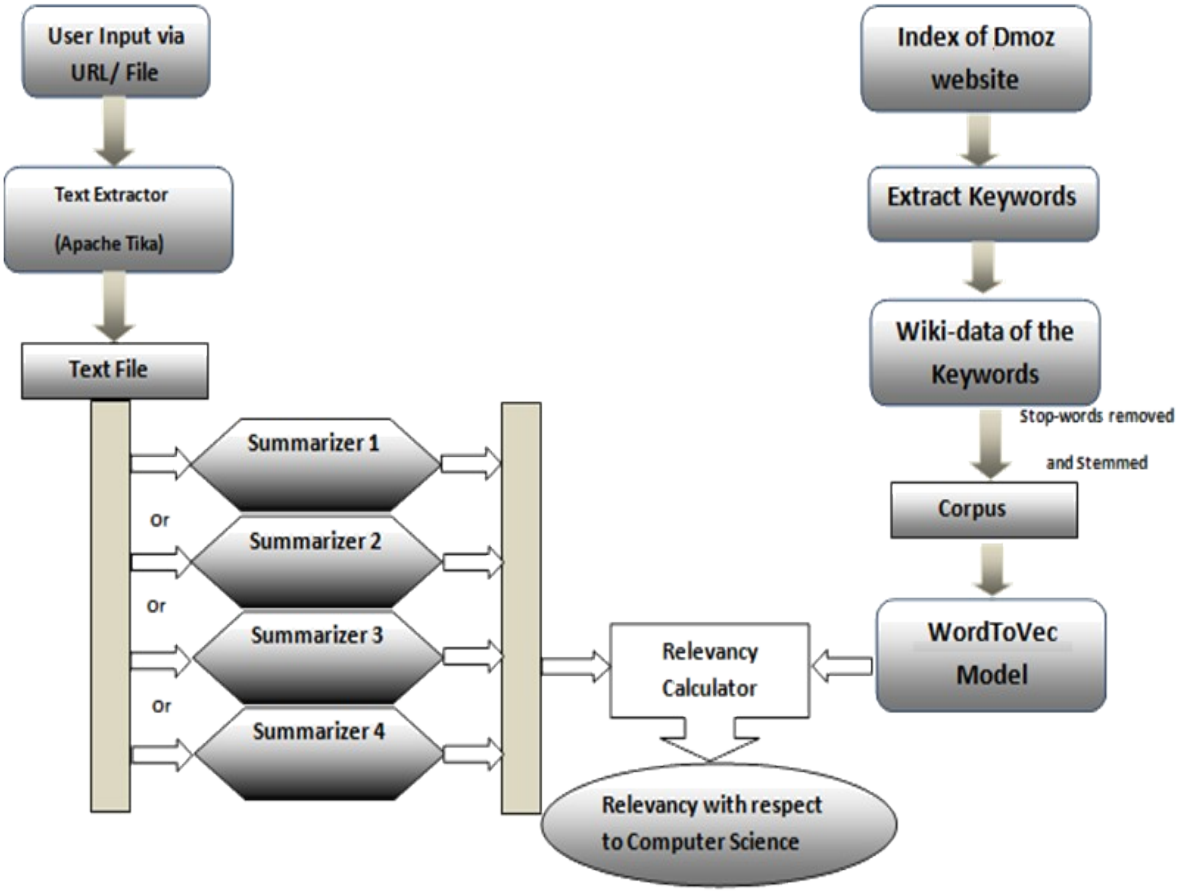
related data corpus and evaluate the summary using similarity between data model and the summary created.

- Crawl the data and create a corpus of related to **Computer Science** domain and create a model using WordToVec tool.
- Given a URL/file, extract the textual content and create a summary using different summarizers.
- Pass the summaries one by one to the WordToVec model and get the relevancy of the summaries with respect to computer science.

4.1) Assumptions:

- The input file/URL will contain valid text format.
- The URL must be a valid one on which the server takes the http requests and responds back.

5) Architecture:



6) Working Model:

The goal of this project is to provide a framework for the developers so that they can further enhance the system using there own summarizers or incorporate the system in their projects to compare the summaries generated by the various summarizers. The whole process of this work is done in 4 different phases which are as follows :-

6.1) Corpus Creation:

- Define a crawler that will crawl through the Dmoz website and get the desired data.

- Get the wikipedia pages of all of these keywords and store them in a text file which is the corpus of our system.
- The wiki pages are being accessed using the Apache-tika tool to get the pages.

6.2) ***Text Extraction:***

- Input for the system can be an URL or any type of file such as pdf, excel, odt, odp etc.These type of files must be converted to text file for the summarizers to manipulate.
- This work is done using Apache-tika tool. Read the input from either the URL or the file, pass it to Apache-tika API and collect the output stream and write it to a file.

6.3) ***Summarization:***

Four Different Summarizers were used to generate the summary for each parsed text document/URL. These summarizers have their own algorithm implemented and they take there own time complexity to generate an efficiently nurtured summary of having high relevance to the developer which can be of high usage. The different summarizers incorporated in the system are as follows :-

- **Summarizer 1** : This Summarizer simply tokenizes the given document and splits it into sentences. Then, it calculates the rank of each sentence according to the TF-IDF Model.
- **Summarizer 2** : This Summarizer is similar to the previous one but has a “min” and a “max” threshold. So, only those sentences are considered which lie in that range.
- **Summarizer 3** : In these summarizers, there is an inbuilt tokenizer and stemmer, uses help of nltk to rank the final sentences.
- **Summarizer 4** : This summarizer is the “Open Text Summarizer”. This summarizer gives us the best relevant results based on the summary ratio we provide to it as input.

There are a available set of summarizers added to the system and more summarizers can be added to the framework. User chooses among the available summarizers and generate the summary. These summaries are being forwarded to the model for relevancy calculation

6.4) Relevancy Calculation:

- The input to the model is the textual summary from all the summarizers. Pass the summary one by one to the model.
- Based on certain parameters the model gives the relevancy factor as the output to all the summaries. These parameters are the similarity of words in the summary to the corpus.
- Based on this factor the user decides, which summary suits the most to the domain.

In this way the whole process of getting the best summaries is processed which involves different stages incorporating different types of tools.

7) Experiments and Results:

The different types of summarizers have their own summary implementation hence their results also varies. The different types of inputs and summarizers that were tested are as follows:-

- The different pages of wikipedia were given as input to all the summarizers and their relevancy factor was compared. This was done with different types of pages.
- The inputs for example was suppose the wiki page "computer_algorithm", the relevancy results were relatively high as the data model was stemmed and no contained no stop words.
- Whereas, for a wiki page called "Gujrat riots" the relevancy when passed to the system was relatively low as the data model was being made of corpus that belonged to computer science domain.

Therefore with the inputs(File or URL) that were related to computer science was much more relevant than an input that did not belong to this domain. This helped the developer in differentiating between different domains.

8) Applications:

- News Feed (Relevancy based on searched category) which means analysing the news and displaying only the summary of the news rather than displaying the whole content.
- Developed as a platform for the researchers working on summarization as they can add new features to this project.

9) Conclusion:

- The project has been developed as a platform into which new summarizers can easily be added.
- Ease for developers to decide which summarizer works best for their domain by testing their data on the summaries and calculating the relevance factor.
- Input any type of file or URL to the platform.

10) Future Work:

Lot of features can be added to the current system such as:-

- The current system lacks the GUI. A much better GUI can be made for the system so that the future developers can directly access the interface to interact with the system.
- The project is deliverable at the framework level so the developers can incorporate it in their projects. They can even add more summarizers to the system for their own convenience.

11) References:

- ***Open Url Directory For Computer Science***

  (http://www.dmoz.org/Computers/Computer_Science

- ***WORD2VEC model***

  Link: http://radimrehurek.com/gensim/index.html

- ***Summarizers***

- http://glowingpython.blogspot.in/2014/09/text-summarization-with-nltk.html
- https://pypi.python.org/pypi/sumy/0.3.0
- http://pythonwise.blogspot.in/2008/01/simple-text-summarizer.html