TEXT SUMMARIZER

MACHINE LEARNING

SLOT – E2

Submitted by:

15BCE0235 – NIKHIL GUPTA

Under the Guidance of:

Prof. BALAKRISHNAN P

Associate Professor

SCOPE

INDEX

ABSTRACT

There is a large amount of textual content, and it is only growing every single day. Think of the internet, comprised of web pages, news articles, status updates, blogs and so much more. The data is unstructured and the best that we can do to navigate it is to use search and skim the results. Text summarization is a process of extracting or collecting important information from original text and presents that information in the form of summary. Text summarization has become the necessity of many applications for example search engine, business analysis, market review. Summarization helps to gain required information in less time. This project is an attempt to summarize and present the view of text summarization from every aspect from its beginning till date.

In this project we address the automatic content summarization task. Recent research works on extractive-summary generation employ some heuristics, but few works indicate how to select the relevant features. We will present a content summarization procedure based on the application of trainable Machine Learning algorithms which employs a set of features extracted directly from the original text. These features are of two kinds: statistical – based on the frequency of some elements in the text; and linguistic – extracted from a simplified argumentative structure of the text.

.

INTRODUCTION

Automatic text processing is a research field that is currently extremely active. One important task in this field is automatic summarization, which consists of reducing the size of a text while preserving its information content. A summarizer is a system that produces a condensed representation of its input text. Summary construction is, in general, a complex task which ideally would involve deep natural language processing capacities. In order to simplify the problem, current research is focused on extractive-summary generation . An extractive summary is simply a subset of the sentences of the original text. These summaries do not guarantee a good narrative coherence, but they can conveniently represent an approximate content of the text for relevance judgement.

A summary can be employed in an indicative way or in an informative way – to cover all relevant information of the text. In both cases the most important advantage of using a summary is its reduced reading time. Summary generation by an automatic procedure has also other advantages:

(i) the size of the summary can be controlled

(ii) its content is determinist

(iii) the link between a text element in the summary and its position in the original text can be easily established. We deal with an automatic trainable summarization procedure based on the application of machine learning techniques. Projects involving extractive summary generation have shown that the success of this task depends strongly on the use of heuristics, unfortunately few indicatives are given of how to choose the relevant features for this task. We will employ here statistical and linguistic features, extracted directly and automatically from the original text.

LITERATURE SURVEY

In this section, we explain existing TS systems that produce summaries automatically, as well as previous work on different approaches that have used TS techniques to address specific tasks (e.g. for generating Wikipedia articles, weather forecast reports, etc.).

Regarding the TS systems, we can find a wide range of them for generating different types of summaries. One of the most cited summarizers is MEAD. This system produces extractive single- and multi-document generic or query-focused summaries in English and Chinese, thus being also a multilingual summarizer. For determining the most important sentences, it relies on the calculation of surface features, such as sentence position, sentence length, similarity with the first sentence, and similarity with a centroid, which are then combined linearly. SUMMA follows a similar approach, also relying on the combination of statistical, positional and similarity features. In particular, the main features are: i) sentence similarity to the centroid; ii) sentence position and iii) similarity with the lead part of the document where the sentence comes from. It uses a vector space representation, where each vector position contains the term and its tf-idf, and it is capable of producing single- and multi-document extractive summaries, as well as generic or query-focused summaries. QCS system integrates a summarization approach within a broad process for retrieving and clustering information. For generating summaries from a set of documents, it first generates single-document summaries, and then a second summarization process is applied to these summaries in order to obtain the final one. For detecting relevant sentences, Hidden Markov Models are employed after some sentences have been trimmed. The sentences with highest probability in the model are chosen for the summary. The AZOM text summarization system for Persian combines statistical and conceptual properties of unstructured documents and generates a summary with a specific structure. Other systems, such as SummGraph, employ graph-based algorithms and knowledge databases for detecting relevant content in the documents. In particular, this system has been proven to work successfully in the newswire, biomedical, and tourist domain, and the techniques used have been also successfully employed for retrieving information from medical records.

Furthermore, we can also find systems that are especially targeted to produce a specific type of summaries (e.g., sentiment-based summaries) or deal with multi-linguality. That is the case of CBSEAS which generate multi-document extractive sentiment-based summaries, or MUSE — MUltilingual Sentence Extractor , that employs language-independent techniques for generating summaries in English and Hebrew.

In other contexts, TS techniques and approaches have been used for solving specific tasks. For instance, Balahur and Montoyo use opinion mining techniques for extracting opinion features from customer reviews and then summarizing them. Sauper and Barzilay propose an automatic method to generate Wikipedia articles, where specific topic templates, as well as the information to select are learnt using machine learning algorithms. The templates are obtained by means of recurrent patterns for each type of document and domain. For extracting the relevant content, candidate fragments are ranked according to how representative they are with respect to each topic of the template. Other approaches that also rely on the use of templates to organize and structure the information previously identified, are based on information extraction systems. In Kumar et al, reports of events are generated from the information of different domains (biomedical, sports, etc.) that is stored in databases. In such research, human-written abstracts are used, on the one hand, to determine the information to include in a summary, and on the other hand, to generate templates. Then, the patterns to fill these templates in are identified in the source texts. Similarly, in Carenini and Cheung, patterns in the text are also identified, but since their aim is to generate contrastive summaries, discourse markers indicating contrast such as "although", "however", etc. are also added to make the summary sound more naturally. To generate summaries from the complete biography of a person is also interesting. This can also be considered a particular type of multidocument summarization, since its goal is to produce a piece of text containing the most relevant aspects of a specific person. Instead of using templates for detecting which information should be of interest from a person, Zhou et al. analyzed several machine learning algorithms (Naïve Bayes, Support Vector Machines, and Decision Trees) to classify sentences, distinguishing between those ones containing biographic information (e.g. the date/place of birth) from others that do not.

Natural Language Generation (NLG) has been also applied for adding new vocabulary and language structures in summaries. In Yu et al. very short summaries are produced from large collections of numerical data. The data is presented in the form of tables, and new text is generated for describing the facts that such data represent. Belz also suggests a TS approach based on NLG, in order to generate weather forecast reports automatically.

EXISTING METHODOLOGY

One example that might come readily to mind is to create a concise summary of a long news article, but there are many more cases of text summaries that we may come across every day.

Extractive text summarization involves the selection of phrases and sentences from the source document to make up the new summary. Techniques involve ranking the relevance of phrases in order to choose only those most relevant to the meaning of the source.

Abstractive text summarization involves generating entirely new phrases and sentences to capture the meaning of the source document. This is a more challenging approach, but is also the approach ultimately used by humans. Classical methods operate by selecting and compressing content from the source document.

This process of content summarization is divided into 3 steps -

Pre-processing – In this structured representation of original content is obtained .

Processing – algorithm is applied to transform Content structure into summary.

Generation – final summary is obtained.

There are two main approaches to summarizing documents; they are:

In the first approach the aim of the pre-processing step is to reduce the dimensionality of the representation space, and it normally includes:

1. stop-word elimination – common words with no semantics and which do not aggregate relevant information to the task (e.g., “the”, “a”) are eliminated;

2. case folding - consists of converting all the characters to the same kind of letter case - either upper case or lower case;

3. stemming - syntactically-similar words, such as plurals, verbal variations, etc. are considered similar; the purpose of this procedure is to obtain the stem or radix of each word, which emphasize its semantics.

A frequently employed text model is the vectorial model.

After the pre-processing step each text element – a sentence in the case of text summarization – is considered as a N-dimensional vector. So it is possible to use some metric in this space to measure similarity between text elements. The most employed metric is the cosine measure, defined as cos θ = () / (|x| . |y|) for vectors x and y, where () indicates the scalar product, and |x| indicates the module of x. Therefore maximum similarity corresponds to cos θ = 1, whereas cos θ = 0 indicates total discrepancy between the text elements.

The idea of a "reference summary" is important, because if we consider its existence we can objectively evaluate the performance of automatic summary generation procedures using the classical Information Retrieval (IR) precision and recall measures.

If each original text contains an author-provided summary, the corresponding size-K reference extractive summary consists of the K most similar sentences to the author-provided summary, according to the cosine measure. Using this approach it is easy to obtain reference summaries, even for big document collections

Vuforia SDK consists of two techniques: image file recognition and text content recognition. The paper is based on image recognition. It is provided with image database of two types: cloud database and devices database. The key or one of the highlighted difference between them is that the later does not require intermediate like network connection and its time response is also faster. Targets are limited 100 targets per download and device target database. It has no meta-data support whereas the former requires network connection and cloud recognition.

IMPLEMENTATION OF THE ALGORITHM

1. Importing the required libraries like: TextBlob, numpy and pandas. TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. pandas is a NumFOCUS sponsored project. This will help ensure the success of development of pandas as a world-class open-source project, and makes it possible to donate to the project.

2. Define a function to remove duplicates.

3. Open text file and read text from it.

4. Append sentences found in file to list.

5. Create dataset of important keywords related to domain of article to be summarized.

6. Import data set.

7. Check for the keywords in document and list.

8. Check if keyword is present in the sentences. If yes append, and if no move to next.

9. Check for sentiment subjectivity and set threshold to 0.1. If greater append the strings to list.

10. Print the sentences.

DESCRIPTION OF THE DATASET

For our experiments, a set of 12 technology related articles from a Google was collected directly from the Web.

Specifically, these articles belonged to the technology and we will be summarizing article related to technology.1

The structure is similar for all of the articles in the corpus: title, authors, content, outline and a variable number of sections (the content of these section is what we considered as the text of the article), depending on each article. In addition, each article may contain figures and tables, which are not taken into consideration for generating the summaries.

CONCLUSION

Text summarization is growing as sub – branch of NLP as the demand for compressive, meaningful, abstract of topic due to large amount of information available on net. Precise information helps to search more effectively and efficiently. Thus text summarization is need and used by business analyst, marketing executive, development, researchers, government organizations, students and teachers also. It is seen that executive requires summarization so that in a limited time required information can be processed. This paper takes into all about the details of both the extractive and abstractive approaches along with the techniques used, its performance achieved, along with advantages and disadvantages of each approach. Text summarization has its importance in both commercial as well as research community. As abstractive summarization requires more learning and reasoning, it is bit complex then extractive approach but, abstractive summarization provides more meaningful and appropriate summary compare to extractive. Through the study it is also observed that very less work is done using abstractive methods on Indian languages, there is a lot of scope for exploring such methods for more appropriate summarization.

REFERENCES

1. Saranyamol C S, Sindhu L, “A Survey on Automatic Text Summarization”, International Journal of Computer Science and Information Technologies, 2014,Vol. 5 Issue 6.

2. Reeve Lawrence H., Han Hyoil, Nagori Saya V., Yang Jonathan C., Schwimmer Tamara A.,

Brooks Ari D., “Concept Frequency Distribution in Biomedical Text Summarization”, ACM 15th Conference on Information and Knowledge Management (CIKM), Arlington, VA, USA,2006.

3. Blog.mashape.com/list-of-30-summarizer-apis-libraries-and-software.

4. Khan Atif, Salim Naomie, “A review on abstractive summarization Methods”, Journal of Theoretical and Applied Information Technology, 2014, Vol. 59 No. 1.

5. Suneetha Manne, Zaheer Parvez Shaik Mohd. , Dr. S. Sameen Fatima, “Extraction Based

Automatic Text Summarization System with HMM Tagger”, Proceedings of the International Conference on Information Systems Design and Intelligent Applications, 2012, Vol. 132, P.P 421-428.

6. Ragunath R. And Sivaranjani N., “Ontology Based Text Document Summarization System

Using Concept Terms”, ARPN Journal Of Engineering And Applied Sciences, 2015, Vol. 10,

No. 6. 25. Patil Pallavi D, Mane P M, “A Comprehensive Review on Fuzzy Logic & Latent

Semantic Analysis Techniques for Improving the Performance of Text Summarization”, International Journal of Advance Research in Computer Science and Management Studies, 2014, Vol. 2, Issue 11, pg. 476-485.

7. Dixit Rucha S., Apte S. S., “Improvement Of Text Summarization Using Fuzzy Logic Based Method”, IOSR Journal Of Computer Engineering (IOSRJCE) ISSN: 2278-0661, ISBN: 22788727,2012, Vol. 5, Issue 6, PP 05-10.

8. Fachrurrozi M., Yusliani Novi, and Yoanita Rizky Utami, “Frequent Term based Text Summarization for Bahasa Indonesia”, International Conference on Innovations in Engineering and Technology Bangkok (Thailand), 2013.

9. Babar S.A. and Thorat S.A., “Improving Text Summarization using Fuzzy Logic & Latent

Semantic Analysis”, International Journal of Innovative Research in Advanced Engineering

(IJIRAE), 2014, Vol. 1 Issue 4.29. Megala S. Santhana, Kavitha