Text - Summarization using Conceptual Statistical Extraction

Vivek Patani
School of Informatics and Computing
Indiana University
Bloomington, Indiana – 47405
Email: vpatani@umail.iu.edu

Jinsu Kim
School of Informatics and Computing
Indiana University
Bloomington, Indiana – 47405
Email: vpatani@umail.iu.edu

Abstract—**There have been increasing number of researches conducted in the domain of text summarization using different types of contexts using different methods. In this paper, we attempt to summarize text with important facts eliminating redundant or irrelevant information using extractive method. In this paper we focus mostly on extracting entities and concepts which occur more commonly and are meaningful. Our approach is a statistical one, wherein we weigh each concept in terms of relations and how frequently they occur.**

I. Introduction

With increasing amount of textual information available, it becomes difficult to find and read important text. Thus, text summarization would be very useful in that it produces important and concise summary and help readers to save time without having to read entire text.

Summary can be generated by **extractive** and **abstractive** methods. Extractive methods create short summary with each words, POS from whole sentences without modification of words. Abstractive summarization is more complex way, which makes new sentences summarize by paraphrasing sentences of source documents.

The paper talks about the selective picking up of entities in the text. We, in this paper weigh **entities** heavily over other concepts. An entity may be a person or any proper noun or moreover any noun. We along with this also collect the verbs associated to them. However, a deep dive into the subject tells us that not all verbs represent an important concept related to the paragraph, hence the term selective.
We divide the approach into **5** sections:

- Sentencifying
- Anaphora Resolution
- Dependency Parsing
- Statistical Inference
- Entity Recognition

We also use a lot of other ideas, but this briefly describes a good overview of the subject we are presenting here. Our focus is on English for different languages hold different semantic syntax, such like languages like Japanese and Chinese have unambiguous sentence ending markers.

II. Approach and Methodology

A. Sentencifying

To begin with we approach the data with splitting of the set of text into simple sentences. We do this to make relations and dependencies much easier and flexible to detect. The idea here is to break the sentence down at periods with a few reservations such as places like *Dr., Mr., St., etc.* 47% of the articles written in the Wall street journal represent abbreveations[2]. This problem in NLP is also referred to as Sentence Boundary Disambiguation (SBD), which is the science of detection of the begining and the ending of a sentence.

B. Anaphora Resolution

The problem of Anaphora Resolution is also termed as pronoun resolution. Let us understand this problem with a brief example, Sentence: Tim walks his dog, he does so every single day. Your mind subconciously decodes that in this context he is Tim and not the dog. The computer not so intelligent as you, needs a few instruction. There are two approaches to solve this problem, namely Eliminative Constraints & Weighing Preferences. The former uses a set of rules which should be obeyed by both the anaphor and referent and latter method weighs each of the eligible referents and picks the one with the highest score[3].

C. Dependency Parsing

This is the most important step in summarizing text. Dependency grammar is based on the relationships between words. $A \to B$, means B is dependent on A or A governs B. Dependency Parser is simply used to realise the predicate, subject and object in a particular context. We can understand a simple selection of dependency by expressing the structure in head – dependent relation, functional categories and some structural categories. Phrases can also be represented by simply the same way. Here is an example of a sentence provided with their dependencies parsed in *fig 1.*

D. Statistical Inference

Once we process the dependencies, we lucidly look for the most important term in all of the text. The term m̈ost importantc̈an be interpreted in many ways. For us, we simply

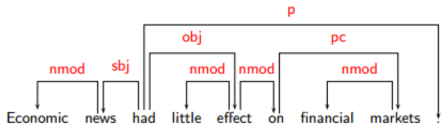

Fig. 1. Dependency Relation

pick the term that has occured in most number of concepts (A concept is a tuple of (predicate, subject, object). We pick that term and look for important concepts then and then in those concepts we look for relevant concepts. The advantage of this is that we are looking for the best possible term and finding all the things related to it and pick the best one. The downside is that once a *best term* is picked, only relevant concepts are searched, so our scope is limited to a single *best term* and can cause a single point of failure. We also look for the best predicates, subjects and objects to find out more information in regards to the *best term*.

E. Entity Recognition

This is the last part and is optional. Named Entity Recognition (NER) is a very interesting property to know for a certain concept or term. It helps us realise whether if the term or concept is a person and often times also helps us realise the gender. This can be helpful in the selection of templates as we can have a customised one for concepts representing people.

III. Practical Implementation

We have implemented the above mentioned idea on Github. You can access the readme and follow the details to run the project.
Before you begin, we need to satisfy a few prerequisites, such as:

- Python 3.5+
- Java 1.8+
- MariaDB 10+ (Any MySQL running on default port should do)
- Docker

Let's understand how it works:

1) Data to be processed in present in a file and the file is read as input (UTF-8) format.
2) Primarily we break down the text into sentences by running it through a sentencer (Stanford NLP provides this).
3) We then take each sentence and run it through a coreref relation generator. This is done to resolve any anaphoras that come across.
4) Once we complete that we take each anaphora and replace each of those with the actual entity.
5) Following that we push it through a Neural Net Depependency Parser (using Stanford NLP). This is the most important step and will cause your quality up or down depending upon how complex sentence structure is.
6) Followed by that we save all the data in a MariaDB database. We store it in a triple store form (predicate, subject, object).
7) We then perform statistical inferences based on the data stored.
8) We find the best term, frequency of each predicate, subject and object and then look for concepts accordingly.
9) Best individual concept and best individual term is giving more prefernce as compared to other things.
10) The best term decides the next step, i.e. Templating.
11) The best term is processed through a Named Entity Recognition Tool.
12) That decides upon what template to pick up and finally that is template we display our best results in.

IV. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
[2] Texting Mine Online
[3] Stanford NLP Text
[4] Sandra Kubler and Marcus Dickinson