



# INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume 5, Issue 3)

Available online at: [www.ijariit.com](http://www.ijariit.com)

## Automatic text summarization

Anagha Shamprasad  
[anaghasham97@gmail.com](mailto:anaghasham97@gmail.com)  
Sai Vidya Institute of Technology,  
Bengaluru, Karnataka

Akshatha G Krishna  
[akshathagk.15cs@saividya.ac.in](mailto:akshathagk.15cs@saividya.ac.in)  
Sai Vidya Institute of Technology,  
Bengaluru, Karnataka

B M Shashank  
[shashankbm.15cs@saividya.ac.in](mailto:shashankbm.15cs@saividya.ac.in)  
Sai Vidya Institute of Technology,  
Bengaluru, Karnataka

Reshma J  
[reshmaj.15cs@saividya.ac.in](mailto:reshmaj.15cs@saividya.ac.in)  
Sai Vidya Institute of Technology,  
Bengaluru, Karnataka

Lokesh S  
[lokesh.s@saividya.ac.in](mailto:lokesh.s@saividya.ac.in)  
Sai Vidya Institute of Technology,  
Bengaluru, Karnataka

### ABSTRACT

*In the recent years we have seen an increasing amount of data generation in every field. It becomes our utmost priority to manage this data. The data produced can be structured or unstructured. When it comes to gain knowledge through data, it becomes easier if we only get a summary out of it. In this digital era, most of us do not have enough time to go through many pages of data. Text Summarization aims to retrieve only the important information out of a large text document. This can be achieved through NLP techniques.*

**Keywords**— Natural Language Processing (NLP), Extractive summarization and abstractive summarization, Entity recognition, Relationship extraction, Template design, Summary generation

### 1. INTRODUCTION

With the rapid growth of the Internet, people are overwhelmed by the tremendous amount of online information [1]. We need Automatic Text Summarization for the following 6 reasons [2].

- Reading time is reduced.
- The selection process of research documents becomes easier.
- The effectiveness of indexing will be improved.
- We obtain unbiased summaries through summarization algorithms.
- We can obtain personalized information.
- The various summarization techniques enable commercial services of providing summaries based on word count.

Natural Language Processing (NLP) is a method that interprets human language from one structure to another. Summarization comes under NLP as a research work. It mainly focuses on providing relevant summary using various Natural Language Processing tools and techniques [3]. In order to deal with the great amount of information across the digital world, there is a need to have an automatic summarization method. Basically, Summarization is categorized into two types - Extractive and Abstractive [4] [5]. Extractive summarization is a method of

summarization which uses the existing sentences and produces a summary out of that. Whereas Abstractive summarization will extract the main keywords and forms meaningful sentences. The goal of this method is to select whole sentences (without modifying them) to create a short paragraph summary.

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.

### 2. EXISTING MODEL

When we want to shorten a text document with software in order to create a summary with the major points of the original document, there are mainly 4 phases – Entity Recognition, Relationship Extraction, Subgraph Construction, Template Design and finally Summary generation.

#### 2.1 Entity Recognition

It is a sub-task of information extraction that seeks to locate and classify the text into pre-defined categories such as the names of persons, organizations, locations, quantities, monetary values, percentages etc., where each of it becomes an entity.

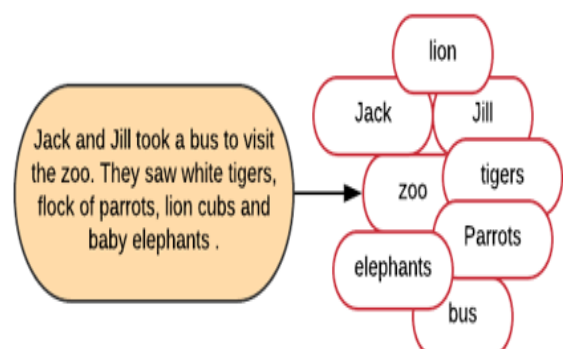
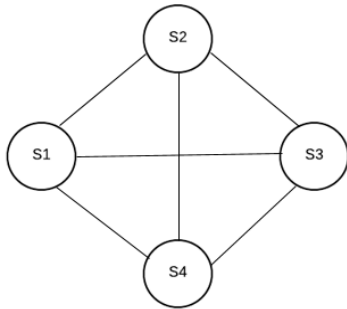


Fig. 1: Entities extracted from a document

### 2.2 Subgraph Construction

After the entities are extracted through First stage method the document should be computed to extract key sentences, which are still in the original document order. To handle the problem of key sentence extraction, we utilizes a graph model. A subgraph is built from a single document, or we can build sub graph from multiple documents. Therefore, the use of knowledge-based graphs can be a good way to complete the task of multi-text summary.



- Node -> Sentence
- Edge -> Similarity between 2 sentences

Fig. 2: A general notation of a subgraph

### 2.3 Template Design

The extraction summary template we propose is accomplished by combining different SPARQL query languages for knowledge graph [23]. Among them, the simplest template traverses all the directional triplets of the document subgraph into the syntactic structure of the subject-predicate-object, and simply combines these sentences to form a summary. The so called complex template is adding the tasks of knowledge inference and aggregation of the subgraph, and accomplishes similar human reasoning. Table 1 shows the difference between the simple and complex templates.

Table 1: Examples of choosing a simple and a complex template

Simple Template	Jack see animals. Jill see animals
Complex Template	Jack and Jill visit the zoo and see animals and birds

### 2.4 Summary Generation

Table 2: Choosing a query template

Since the information extracted from the text is the structure of the graph, the abstract of the text can be extracted by asking questions. A sample of choosing the questions to produce a summary is shown in the table below.	Who visited the Zoo? What did they see?
Question Summary	Jack and Jill visited the zoo. They saw animals

## 3. PROPOSED SYSTEM

In the existing system, after the generation of a summary for a document of say about 1000 words, the summary is cut down to almost about 800 words. To further shorten the summary and to acquire more precision, the proposed model includes another phase other than the 4 main phases. This phase is known as the Relationship Extraction Phase.

We establish a candidate relationship set between the existing entities. Then, the relationship of the subject-predicate-object

structure is determined by the syntactic dependency tree, and the relationship pair with the highest probability is chosen. Words like located in, employed by, part of, married to etc all represent relationship among entities. Based on the availability of training data and annotated text we can perform relation extraction in 5 methods:

- (a) Hand-built patterns
- (b) Bootstrapping methods
- (c) Supervised methods
- (d) Distant supervision
- (e) Unsupervised methods

### 3.1 Hand-built patterns

Hand built rule-based systems have a naive approach towards relation extraction task. They come up with the relations between entities by analyzing set of sample examples and writing a possible set of rules which obey it. For example, Agar is a substance prepared from a mixture of red algae, such as Gonidium, for laboratory or industrial use. While reading this sentence human can predict, there is a hyponym relation between red algae and Gonidium. We predict this by observing the connecting words 'such as' between these two words. For identifying relations such as hyponyms we can use this technique.

### 3.2 Bootstrapping Method

If we don't have enough annotated data to train and lots and lots of unannotated text for relation extraction then, we will not get a good result. The solution for this approach is bootstrapping technique. In this approach, we have some seed instances, which is manually tagged data used for the first phase of training called the seed instances. We train with seed instances and learn the classifier, and test with the classifier, and get more train examples by adding the test results to the training set. Thus, the training set will grow up to a sufficient amount. This approach can be called as a semi-supervised model. [6].

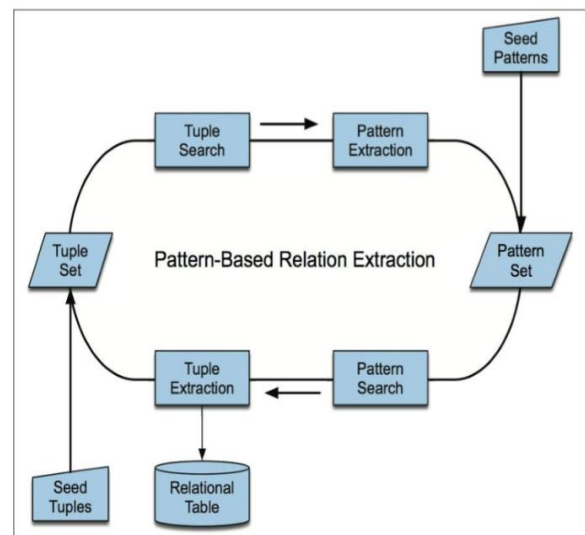


Fig. 3: Bootstrap Relation Diagram

### 3.3 Supervised Methods

Supervised systems are the state-of-the-art system for many natural language processing tasks. The supervised system will learn from the already tagged corpus with the help of features or learn by the systems. The key idea for the supervised learning is to model the relation extraction task as a classification problem (Binary, or multi-class) and train the classifier with different techniques for prediction of new relations. We take help from various available machine learning algorithm for designing the classification problem. For the sake of simplicity and clarity, we

restrict our discussion to binary relations between two entities.

Given a sentence

$S = w_1, w_2, \dots, e_1, \dots, w_j, \dots, e_2, \dots, w_n$ ,

Where  $e_1$  and  $e_2$  are the entities, then a mapping function  $f(.)$  can be given as:

$$f_R(T(S)) = \begin{cases} True, & \text{if } e_1 \text{ and } e_2 \text{ are related according to relation } R \\ False, & \text{Otherwise} \end{cases}$$

Where  $T(S)$  are the features extracted from the sentence  $S$ . The mapping function  $f(.)$  defines whether there exists a relation between these entities. [6]

### 3.4 Distant Supervision

Modern models of relation extractions are based on supervised learning of relations from small hand-labeled corpora. An alternative paradigm that does not require labeled corpora, and allowing the use of corpora of any size is distant Supervision which combines both supervised and unsupervised training techniques. This uses Freebase, a large semantic database of several thousand relations. For each pair of entities that appears in some Freebase relation, we find all sentences containing those entities in a large unlabeled corpus and extract textual features to train a relation classifier. [7]

### 3.5 Unsupervised Methods

Unsupervised information extraction, extracts strings of words between entities in large amounts of text, and clusters and simplifies these word strings to produce relation-strings. Unsupervised approaches can use very large amounts of data and extract very large numbers of relations, but the resulting relations may not be easy to map to relations needed for a particular knowledge base.[7]

## 4. METHODOLOGY

This model builds named entity recognizer with NLTK (Natural Language Tool Kit) and SpaCy, to identify the names of things, such as persons, organizations, or locations in the raw text. When assigning weights for words, we can think of binary (0 or 1) or real-value (continuous) weights and decide which words are more correlated to the topic. The two most common techniques in this category are: Word Probability and TFIDF (Term Frequency Inverse Document Frequency).

### 4.1 Word Probability

The simplest method is to use frequency of words as indicators of importance. This is word probability. The probability of a word is determined as the number of occurrences of the word,  $f(w)$ , divided by the number of all words in the input (which can be a single document or multiple documents):

$$P(w) = \frac{f(w)}{N}$$

### 4.2 TFIDF (Term Frequency Inverse Document Frequency)

Since word probability techniques depend on a stop word list in order to not consider them in the summary and because deciding which words to put in the stop list is not very straight forward,

there is a need for more advanced techniques. One of the more advanced and very typical methods to give weight to words is TFIDF. This weighting technique assesses the importance of words and identifies very common words (that should be omitted from consideration) in the document by giving low weights to words appearing in most documents. The weight of each word  $w$  in document  $d$  is computed as follows:

$$q(w) = f_d(w) * \log \frac{|D|}{f_D(w)}$$

Where,  $f_d(w)$  is term frequency of word  $w$  in the document  $d$ ,  $f_D(w)$  is the number of documents that contain word  $w$  and  $|D|$  is the number of documents in the collection  $D$ .

## 5. CONCLUSION

Generating abstractive summary is becoming a necessity in this large digital world where huge amount of information is available to get a concise and short summary. It is difficult for humans to summarize large amounts of text. In this paper, we emphasized various extractive and abstractive approaches for single document summarization. We have described some of the most extensively used methods such as Bootstrapping, supervised and unsupervised methods. Although it is not feasible to explain all diverse algorithms and approaches comprehensively in this paper, we think it provides a good insight into recent trends and progresses in automatic summarization methods and describes the state of the art in this research area.

## 6. ACKNOWLEDGEMENTS

The completion of this project brings with a sense of satisfaction, but it is never completed without thanking the person responsible for its successful completion – Prof. Lokesh S, Department of Computer Science & Engineering, Sai Vidya Institute of Technology

## 7. REFERENCES

- [1] Allahyari, Mehdi, et al. "Text Summarization Techniques: A Brief Survey." arXiv preprint arXiv: 1707.02268 2017.
- [2] The book "Automatic Text Summarization" written by Juan-Manuel & Torres-Moreno and published in 2014
- [3] Sunitha C, Dr. A Jaya and Amal Ganesh "Abstractive Summarization Techniques in Indian Languages" Peer-review under responsibility of the Organizing Committee of ICRTCSE 2016 doe: International Conference of recent trends in computer science., 2016
- [4] Khan, Atif, and Naomie Salim. "A review on abstractive summarization methods." Journal of Theoretical and Applied Information Technology 59.1: 64-72, 2014
- [5] Dalal, Vipul, and Latesh G. Malik. "A survey of extractive and abstractive text summarization techniques." Emerging Trends in Engineering and Technology (ICETET), 2013 6th International Conference on. IEEE, 2013.
- [6] <http://www.cfilt.iitb.ac.in/resources/surveys/nandakumar-relation-extraction-2016.pdf>
- [7] <https://web.stanford.edu/~jurafsky/mintz.pdf>