

**POLITECNICO DI TORINO**

**Master's degree in Computer Engineering**

Master Thesis

**Automatic Extraction of Slides from  
Scientific Papers**



**Supervisors**

prof. Luca Cagliero  
prof.ssa Laura Farinetti

**Candidate**

Du Yaxue  
231696

March 2021

# Abstract

Presentation slides have been an important and effective choice to deliver the key information of a scientific paper, especially in conferences. Most of the time, the speakers have to spend a major amount of time preparing for making slides because scientific papers are different to read and summarize the content. Automatic extraction of slides from scientific papers will be quite helpful and time-saving during the preparation of presentation slides by providing the keywords and bullet points.

In this paper, a system is proposed to automatically extracting the presentation slides from scientific papers. The extracted slides can be used as drafts to help the writing of the final presentation. It firstly applies an unsupervised summarization algorithm to rank the sentences in the scientific paper based on relative importance. Then, it selects the most relevant sentences and phrases to include in each slides using an optimization-based algorithm, which is based on the Integer Linear Programming (ILP) Module with a set of sophisticated constraints. Finally the selected key sentences and phrases can be used to output the well-structured slides. The evaluation rouge results is based on the test of 195 paper-slides pairs which is collected from the website.

Our goal is to compare the performance of our implemented unsupervised summarization algorithms with supervised learning algorithms, to verify whether an unsupervised summarization algorithm could achieve a higher rouge score than a supervised algorithm and finally to find a better performance system for automatically extracting presentation slides from scientific papers.

Based on our experimental results, the F-Measure score of ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4 are taken into consideration for evaluating the performance of different tested methods with various parameter settings. Supervised methods outperform the implemented unsupervised methods with obvious advantages. Among all the supervised methods, ensemble methods (Gradient Boosting and Random Forest and Decision Tree) performed better for all the Rouge scores. RFR is always performs best. MLP and SVR performed slightly worse, probably due to model overfitting issues. As for unsupervised approaches, either TextRank or LexRank could be the best performance one among unsupervised methods with certain parameter settings and they have little advantages than each other. LSA is always the worst performance one.

## Acknowledgements

Firstly, I would like to express my deepest appreciation to my supervisors prof. Luca Cagliero and prof.ssa Laura Farinetti. Thanks for supporting me a lot since I started to work with this thesis topic. Especially when I was in trouble with my thesis work, my supervisors have provided me valuable guidance.

In addition, I am very thankful to PhD student Moreno La Quatra. He is also generously for helping me with my thesis work. Thanks a lot for providing precious suggestions for solving the problems and making improvements. Also I would like to thank the researches who share their researches resources online.

Finally, I want to thank my roommate Min and landlord Mario. In the tough year of 2020, they both supported me a lot like a family member. Last but important, I would like to thank all my beloved family for their unconditional love and supports on me. Without their invaluable encouragement and help, I would not insist on studying for my master degree.

# Table of Content

1. Introduction.....	1
2. Related Works.....	4
2.1 Brief introduction.....	4
2.2 Supervised text summarization methodologies.....	4
2.3 Unsupervised text summarization methodologies.....	5
2.4 Slides extraction using summarization approaches.....	6
3. Overview of related technologies.....	8
3.1 Machine Learning.....	8
3.2 Natural Language Processing.....	9
3.3 Text Summarization(Automatization).....	11
4. Proposed Method.....	13
4.1 Overview.....	13
4.2 Datasets.....	14
4.3 Datasets pre-processing.....	15
4.4 Sentence importance assessment.....	16
4.4.1 Unsupervised learning models.....	16
4.4.2 Supervised learning models.....	19
4.5 Slides Generation.....	20
5. Experimental results.....	23
5.1 Experimental setting.....	23
5.2 Evaluation metrics.....	24
5.3 Results and Discussion.....	26
6. Conclusions and future works.....	42

## List of Tables

Table 1	Common research tasks in Natural Language Processing.....	10
Table 2	Parameter and value of tested supervised learning models .....	24
Table 3	Parameter and value of tested unsupervised learning models.....	24
Table 4	The ROUGE-1 F-measure scores obtained.....	26
Table 5	The ROUGE-2 F-measure scores obtained.....	27
Table 6	The ROUGE-L F-measure scores obtained.....	27
Table 7	The ROUGE-SU4 F-measure scores obtained.....	27
Table 8	Rouge-1 results of Lsa .....	28
Table 9	Rouge-2 results of Lsa.....	28
Table 10	Rouge-L results of Lsa.....	28
Table 11	Rouge-SU4 results of Lsa.....	28
Table 12	Rouge-1 results of Lexrank.....	28
Table 13	Rouge-2 results of Lexrank.....	30
Table 14	Rouge-L results of Lexrank.....	32
Table 15	Rouge-SU4 results of Lexrank.....	34
Table 16	Rouge-1 results of Textrank.....	36
Table 17	Rouge-2 results of Textrank.....	37
Table 18	Rouge-L results of Textrank.....	38
Table 19	Rouge-SU4 results of Textrank.....	39
Table 20	Rouge-1 results of supervised learning models.....	40
Table 21	Rouge-2 results of supervised learning models.....	41
Table 22	Rouge-L results of supervised learning models.....	41
Table 23	Rouge-SU4 results of supervised learning models.....	42

## List of Figures

Figure 1	An online tool for making presentation slides automatically.....	8
Figure 2	System architecture with unsupervised learning models.....	13
Figure 3	System architecture with supervised learning models.....	14
Figure 4	Datasets pre-processing.....	16
Figure 5	LexRank algorithm.....	18
Figure 6	The flow of the TextRank algorithm.....	19
Figure 7	Simple example of computing Recall and Precision score.....	25

# 1. Introduction

Nowadays plenty of scientific papers and their corresponding presentation slides are distributed on the website by scholars, conferences speakers, professors and scientific researches. It is pretty valuable to do some analysis on the large amount of paper-slides pairs. For example, we can develop the summarization tool which takes the paper-slides pair as the inputs, implements different algorithms and finally generates a short summary. Also we can evaluate a summarization system with paper-slides pairs by comparing the system generated summary with corresponding human-written slides text using evaluation tool.

Text mining, similar to text analytics, is used for discovering and extracting good-quality information from free or unstructured text. We can understand text mining from three different perspectives : information extraction, data mining and knowledge discovery in databases(KDD) process. Information extraction simply extracts facts from texts. Data mining applies some algorithms and methods to original texts in order to find useful patterns. Knowledge discovery in databases process performs several processing steps iteratively to datasets for extracting useful patterns and usually some processing steps require interactive feedback from a user. Text mining employs a variety of methodologies to process the text, one of the most important among these is Natural Language Processing (NLP) which is a techniques for dealing with data to understand underlying meaning. Natural Language Processing is managing to deal with all complexities of linguistic concepts like grammatical and semantic structure, sentiment analysis, etc. Natural Language Processing contains two parts : Natural Language Understanding and Natural Language Generation, which simulates the human ability to create natural language text e.g. to summarize information or participate in a dialogue. It has been more and more professional over the past ten years, with products such as Siri, Alexa and Google's voice search employing NLP to understand and respond to user requests. Natural Language Processing will play a significant role in the future. Everyday there are billions of text data being generated in social media (Instagram, XiaoHongShu, YouTube etc.), blogs, news publishing platforms, google searches, messages apps (Whatsapp, WeChat, Telegram etc.), forums (Quora, Reddit etc.) and many other channels. We can no longer understand the abundant volumes of text data as well as the highly unstructured data source by using the common approach. That is just what NLP could deal with.

Text summarization is a technique to process original large amount of text into a short summary for quicker understanding. It aims to generate a concise, fluent and not redundant summary including only the vital information outlined in the document. With an enormous amount of textual materials, you have to waste large amount time reading, searching and checking whether the large documents contain the information that you are looking for. Text summarization does help to save time by reduce much of this text data to shorter, focused summaries that capture the salient details. There are two main categories of text summarization method: Extractive and Abstractive methods. Extractive text summarization method is a traditional method developed earlier. The main purpose is to identify the sentences which is the most relevant to the meaning of the original text by ranking the relevance of sentences and then add them to the summary. The summary obtained contains extracted sentences from the source text. Abstractive text summarization method is a more challenging and advanced method. This approach is to recognize the important sections, interpret the context and reproduce a new brief text which covers the most critical information in the original text. It ensures that the core information is conveyed through shortest text possible. The sentences in summary are newly generated instead of the extracted ones from original text. Applying text summarization is useful to reduce reading time, accelerate the

process of researching for information, and increase the amount of information. Today, with such a tremendous volume of data gathering in the digital space, it is necessary to develop machine learning algorithms that can automatically summarize longer texts into a shortly summary and deliver accurate summaries that can fluently express the expected messages.

Slide presentation is a powerful way to deliver key concepts, convey and share information to the audiences not only at professional but also educational meetings. Usually researchers prefers to disseminate their scientific results by publications and research presentations. The latter one has been widely used in almost all the scientific domains. PowerPoint can be used to set up the theme and outline of the presentation slides but it can not help researchers in selecting the content for the slides. So the researchers still have to spend lots of time to prepare the presentation slides. Also, in case of research articles and lengthy discussion papers, it is quite frequently that some vital messages are easily missed out. While, automatically slides generation technique is an intelligent tool with minimum human interference to solve these drawbacks of the traditional way of manually preparing slides. It would not only reduce the presenters' time in preparing presentations by providing the generated draft slides, but it also offers a way to summarize the paper by extracting the key phrases and bullet points from the original paper content. Nowadays, more and more researches are focus on the topic of automatically slides generation. Many different methods have been introduced in order to generate high-quality and well-formatted slides. An approach which is using deep neural networks is presented in [Athar Sefid et al., 2019]. This method makes some improvements in the following ways extracting a more overall list of surface features, considering semantic or meaning of the sentence, and using context around the current sentence to rank the sentences based on the previous work. With the ranked sentences, Integer Linear Programming is used to select salient sentences and finally generate the slides. Bidirectional Encoder Representations from Transformers(BERT) model is used in the proposed system [Shaj et al., 2020] to learn and train the papers and slides. The Google BERT algorithm, a custom module that was released by Google, is used to summarize the research papers. The sentence important scores are predicted by the pre-trained model of BERT. The Python's unpdfex tool is used to extract text from the papers. The generated summary is used by BERT for making slides.

The objectives of this thesis work is to proposed a new automated approach to automatically generate slides of scientific paper based on unsupervised document summarization and compare its performance with that of the supervised models and find out an efficient automatically slides generation method.

Our proposed methodology extends the PPSGen [Yue Hu et al., 2015] which is introduced as a tool for automatically generating presentation slides for academic papers. There are two main steps in PPSGen, one is learning the importance of the sentences in paper, in this step, support vector regression (SVR) methods, a supervised learning model is applied to compute the importance score of each sentences. The other is slides generation, integer linear programming(ILP) method is used to selected key phrases and sentences with predefined constrains and the well-structured slides are generated by choosing and positioning key phrases and sentences. Our work insists on the two key steps, but with step one in order to assign an importance score of each sentences in the paper we are implementing different unsupervised document summarization methodologies. Unsupervised Learning mainly deals with the unlabeled data, as indicated by its name, the users do not need to supervise the model when this machine learning technique is implemented. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected. The examples of unsupervised learning models are Lexrank, Textrank and Lsa. Supervised learning models such as SVR, MLP, RFR etc, analyze the training data and produce an inferred function



which maps an input to an output based on labeled training data consisting of a set of training examples. Each example is a pair consisting of an input object (typically a vector) and a desired output value. With the computed importance score of each sentences Integer Linear Programming(ILP) method is implemented to select the both key phrases and bullet points and to generate the well-formatted slides .

Our evaluation rouge results are based on the test of 195 paper-slides pairs. These paper-slide pairs are mainly collected from the homepage of researches or professors. Results are computed with the summaries generated by testing with both unsupervised learning models and supervised learning models mentioned above. For evaluating the quality of the system generated slides, we use ROUGE toolkit. the F-Measure score of ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4 are mainly considered for evaluating the performance of different implemented methods. The author-written slides of paper is regarded as the reference summary. The system generate slides contents are evaluated with all stop words removed and stemming utilized. ROUGE-1 evaluates a system generated summary by matching its unigrams reference summary. ROUGE-2 refers to overlap of between the system summary and reference summary. ROUGE-SU4 matches unigrams and skip-bigrams of a generated text against reference texts. ROUGE-L measures longest matching sequence of words using LCS which does not require consecutive matches but in-sequence matches that reflect sentence level word order. Our evaluation outputs show that supervised learning models perform better than unsupervised learning models. Among all the other supervised learning models, RFR is the winner among supervised methods. As for unsupervised approaches, either TextRank or LexRank could be the best performance one with little advantage than each other.

## 2. Related Works

### 2.1 Brief introduction

Automatic summarization aims to reduce a large text to a short and precise summary which contains its main points. It is implied to produce summaries of the text with the known type, such as scientific papers, research articles, online news or articles. Many different text summarization methodologies have been proposed in order to generate more accurate summary. There are mainly two group of methodologies mentioned in this paper: supervised text summarization methodologies and unsupervised summarization methodologies.

Automatic slides generation is a little bit challenging and meaningful task than automatic summarization. It assists the presenters to prepare for their presentation slides which offer the important message and main contributes of their researches or speech accurately and clearly. It is nowadays much more popular to be studied. Many researches are interested in the topic of automatic slides generations and aim to provide a system that has evident advantages to produce good quality slides.

### 2.2 Supervised text summarization methodologies

A supervised method which is extended on the extractive approaches is introduced in [Cagliero, L., & La Quatra, M.,2020]. This approach is based on regression techniques which are trained on variety of features extracted from previously annotated articles. The similarity score is computed between article sentences and highlights based on n-gram co-occurrences. It aims at extracting highlights of the articles with missing annotations automatically and making the process of manually annotating new articles easily. The test datasets are the huge collection of articles ranging over different fields.

[Mao, X. et al., 2019] proposed three approaches for extracting single document summary. It combines supervised learning with unsupervised learning. Approach 1 implies supervised model and graph model to score sentences separately, and the final score of sentences is assigned by linear combination of scores. In approach 2 the graph model is considered as an independent feature of the supervised model to compute the importance of sentences. In approach 3, supervised model is applied to score the importance of sentences which is used as a priori value of nodes in the graph model, and finally the sentences score is computed using biased graph model. The results which are based on the data sets of DUC2001 and DUC2002 are evaluated with ROUGE method. The evaluation outputs show that all the three methods have achieved good results, and performs better than the methods of extracting summary only using supervised learning or unsupervised learning.

[Jain, A. et al., 2017] proposed an approach to extract a good set of features followed by neural network for supervised extractive summarization. It is superior than various online extractive text. In this approach the problem is modeled as a binary class classification. Only the sentences which involved in positive class will be included in the summary.

[Collins, E. et al., 2017] introduces a new dataset for summarisation of computer science publications and implements some method to extend it. Both neural statement coding and the traditional used summarisation features are used to develop models on the dataset. The results

show the models which encode sentences and their local and global context performs best, obviously perform better than the baseline method.

## 2.3 Unsupervised text summarization methodologies

Enhanced LSA-based Summarizer named as ELSA is introduced in [Cagliero, L. et al., 2019]. ELSA is a new multilingual summarization algorithm which combines the item-based and LSA-based strategies effectively and it selects the most relevant subset of sentences in quadratic time by exploiting greedy strategy. The sentences which represent the most significant concepts are chose by LSA-based summarizers. Here the concepts are modeled as the combination of single-document terms. And they are derived from a sentence by sentence matrix by using singular value decomposition (SVD). Itemset-based summarizers choose the sentences which contain the largest amount of frequent item sets. The frequent item sets are utilized by ELSA to describe all the potential concepts covered by the analysis document, while LSA reduces the potentially redundant item set to a compact set of unrelated concepts. The sentences that cover the underlying concepts with minimum redundancy are picked by the summarizer. The newly proposed summary algorithm are tested on benchmark multilingual document collections. The results show itemset-based, LSA-based summarizers and most of the other state-of-the-art approaches do not perform better than the proposed method.

LexRank, a graph-based approach for calculating the importance score of sentences for Natural Language Processing is introduced in [Erkan, G. et al., 2004]. The newly proposed method is based on eigenvector centrality in a graph representation of sentences. The sentences graph uses a connectivity matrix based on intra-sentence cosine similarity as the adjacency matrix.

[Tixier, A. et al., 2017] proposed an unsupervised extractive text summarization method. The method produces the summary in a greedy manner for retaining near-optimal performance. It uses the greedy algorithm to the novel coverage of the objective function. The components assign the meaningful score to words by implementing the graph of words representation and the k-core decomposition algorithm.

[Miller, D., 2019] introduces a lecture summarization service which is implemented to automatically extract text summarization on lectures by collecting key phrases and sentences that delivery the main content. The service is a RESTful service developed in Python. The BERT model is used for text embeddings and the K-Means clustering to identify which sentences that best represent the content for summary selection.

Another unsupervised method is developed in [Zheng, Hao et al., 2019], the author believes that large-scale and high-quality training data is not available or could be created for different types of summaries, domains, or languages. The author improves the calculation on node(aka sentence) centrality in two ways: (1) BERT, a kind of new neural learning model is employed to better catch the meaning of the sentence, and (2) The graph is built with directed edges. Because in many cases, the contributions of two nodes to the connection of their corresponding centers may not be equal.

TextRank, a graph-based extractive summarization algorithm can be used for processing the summary of texts. [F. Barrios et al., 2016] proposed three different options to the similarity function for the TextRank algorithm. These options are based on changing the way the distances between sentences are calculated to weight the edges of the graph used for PageRank.

Researches not only contribute to the study on single document summarization using unsupervised summarization methods, but also work hard on multi-document summarization. [S. Lamsiyah et al., 2020] proposed an unsupervised approach which is based on centroid method and sentence embeddings for automatically summarizing a set of documents. This approach takes consideration of a final score which is calculated by the combination of three scores sentence content relevance, sentence novelty, and sentence position scores. The final score indicates whether the relevant sentences is valuable for choosing.

## 2.4 Slides extraction using summarization approaches

Slides presentation has been widely used in professional and education meetings, speeches and so on. It is an powerful way to disseminate the key information. The presenter may express the vital message of their researches or speech accurately. It helps the listeners easily understand. The presenters may use tools like PowerPoint to set up the theme and outline of the presentation slides, but PowerPoint can not help researchers in selecting the content for the slides. So the researchers still have to spend lots of time to write the presentation slides. It is likely that some vital things are missed out especially in case of research articles and lengthy discussion papers. The task of slides generation is then becoming urgent.

Automatically slides generation technique is an intelligent tool. It aims to solve these drawbacks of the traditional way of generating slides manually with minimum human interference. By providing the generated draft slides, it could reduce the presenters' time in preparing presentations. At the same time, it also provide a way to summarize the paper by extracting the key phrases and bullet points from the source paper content.

Early in the year of 2003, [Yasumura et al., 2003] introduced a support system takes a technical paper as a TeX document, the number of slides wanted and keywords of the paper as inputs for making slides from a specialized paper. A TeX document paper will be firstly processed into a document of XML format. The weights of terms such as sentences, figures and tables in the paper are calculated by utilizing TF\*IDF method. The system decides the number of slides are assigned to each section based on the weights. The user can also reassign and rearrange the slides according to their appetite using a presentation editor.

[T. Shibata et al., 2005] a method is proposed to extract summary slides from a text. This method is based on syntactic or case analysis, the topic/non-topic parts that are itemized to finally create slides. The discourse structure identified by cue phrases, identification of word chain and similarity between two sentences is used to control the indentations of the items .

[D. Galanis et al., 2012] presented a new approach to produce extractive multi-document summaries. Integer Linear programming is used to maximize the importance of the sentence . It includes in the summary and their diversity, without exceeding a maximum allowed summary length. A Support Vector Regression model is used to train on human-written summaries in order to assign an importance score for each sentence. The number of distinct word bigrams in resulting summary identifies the diversity of the selected sentences.

Nowadays, more and more researches are worked on automatically slides generation. Many different methods have been introduced in order to generate high-quality and well-formatted slides. Most of the methods are processing with only the textual information, there are also some tools deal with pictures, charts and table and manage to add them in order to make the presentation slides more accurate.

[Yue Hu et al., 2015] the author proposed a unusual system called PPSGen for automatically generating presentation slides for academic papers. PPSGen initially applies regression methods to learn the importance of the sentences in paper, and then the well-formatted slides are generated by the integer linear programming(ILP) method by choosing and positioning key phrased and sentences.

[Autade Dhanshri P et al., 2016] introduced the framework of a system to automatically generate presentation slides which saves the time of the presenter and helps in creating a structured summary the for a technical paper. The input paper is the one has an abstract and with the following sections such as introduction, related work, model, experiments and conclusions. In this system, documents in LATEX are considered as input. These documents are firstly converted to XML format. And then parsing the XML file and extracting the information in it. Finally, a query specific extractive summarizer has been implemented to generate slides. All graphical elements from the paper are made well use by placing them at appropriate locations in the slides. These slides are presented in the document order.

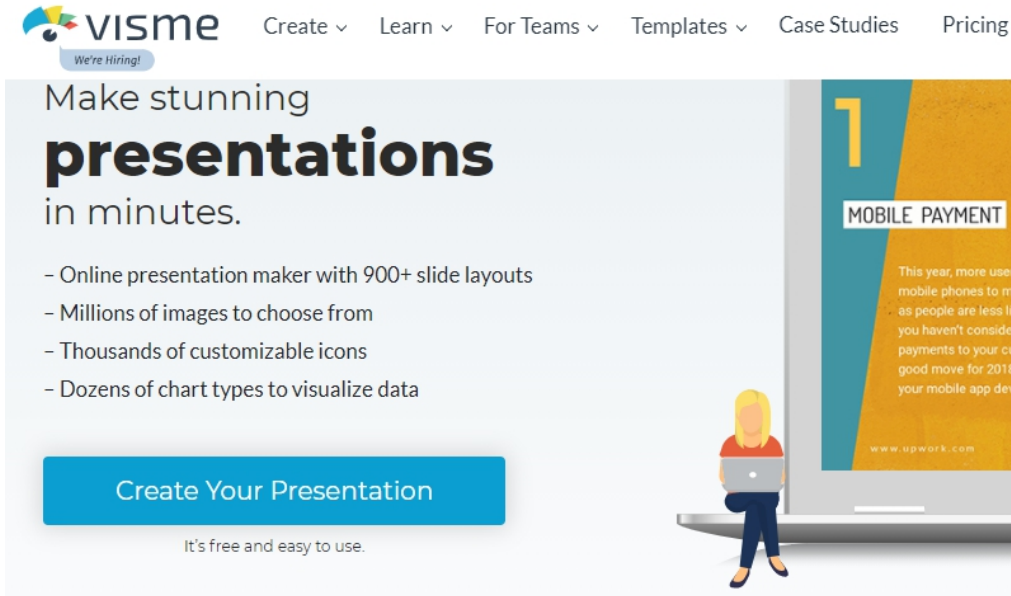
[Athar Sefid et al., 2019] in this paper, an approach which using deep neural networks for automatically generating presentation slides for scientific papers is introduced. Their method makes the following improvements such as 1)extracting a more comprehensive list of surface features, 2)considering semantic or meaning of the sentence, and 3)using context around the current sentence to rank the sentences based on the previous work. With the ranked sentences, Integer Linear Programming is used to select salient sentences and finally the slides are generated with the selected sentences.

[Shaj et al., 2020] the proposed system learns and trains the papers and slides by Bidirectional Encoder Representations from Transformers(BERT) model. The Google BERT algorithm which is a custom module that was released by Google, it is used to summarize the research papers. The sentence important scores are predicted by the pre-trained model of BERT. The Python's unpdf tool is used to extract text from the papers. The generated summary is used by BERT for making slides.

Most recently, [TJ Fu et al.,2021] a new task and method for document-to-slide generation. Different from some previous proposed system, for organizing the key elements in a way suitable for presentation this method takes document summarization, image and text retrieval, slide structure and layout prediction into considerations. In order to solve their task in an end-to-end way, they also propose a hierarchical sequence-to-sequence approach.

You can easily find many online tool with a clear and tidy user interface for generating slide generation automatically. By simply uploading the document you want to present, the tool can output the corresponding presentation slides. It is convenient for users to prepare the slides. But automatically slides generation task is still a challenge task for researches. More and more efforts should be make for a better performance.

The topic of automatic slide generation still has a long way to go. Improvements can be made on the sentences, pictures and figures selection and slides generation which helps to generate well-structured presentation slides.



**Figure 1** An online tool for making presentation slides automatically

## 3. Overview of related technologies

### 3.1 Machine Learning

Machine Learning, a specific part of artificial intelligence(AI), can learn from the massive amounts of historical data, find patterns in it and predict new output results by applying the patterns. It improves the accuracy of software application on predicting outcomes with minimal human intervention.

Machine learning has a long history, statistical methods has been discovered and refined before 1950s. After that it has been developed a lot with many new discoveries and achievements and today's machine learning is no longer like the machine learning of the past in the new computing technologies. In the past, it is derived from pattern recognition and the theory that computers can learn to perform specific tasks without programming. Researchers worked in artificial intelligence fields manage to identify whether computers can learn from data. Because when models are exposed to new data, they can adapt independently. They learn from previous calculations to predict reliable, decisions and results. The machine learning is important and this is not a new science any more, but a new motivation.

Machine learning is classified into four basic categories by how an algorithm learns to be more precious in its predictions, supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of data to be predict determines the type of algorithm a data scientist will use.

#### 1) Supervised learning

It aims to model the correlations between the target prediction output and original input features. We can predict the output values for new data based on those relationships, which it has learned from past datasets. Data scientists choose this algorithm to train the labeled data and assess for correlations of defined variables. Data scientists can specify both the input and the output of the algorithm. The algorithm can compare its predicted

results to the correct expected output to identify errors and improve the model accordingly.

2) Unsupervised learning

It is used to training on not marked or classified data. The algorithm scans through the whole data sets, finds any meaningful connections and draws inferences from the data sets. The trained data algorithms and the expected predicted output are predetermined. Since the model trains with unlabeled data, the algorithms do not have output categories or labels on the data.

3) Semi-supervised learning

Semi-supervised machine learning algorithms involves a mix of the two preceding types supervised learning and unsupervised learning. Both labeled and unlabeled data are used for training purposes. Typically, it is a combination of a small amount of tagged data and a large amount of untagged data. This method can greatly improve the learning accuracy of system. Frequently, when the collected markup data requires skilled and relevant resources to learn from a semi-supervised learning system is the better choice.

4) Reinforcement learning

It is also a popular type of machine learning. The reinforcement learning algorithm learns from its environment continuously using iteration in order to maximize its performance and get the best results. It aims at using observations gathered from the interaction to interact with the environment to generate actions and then find errors or rewards and finally take actions for maximizing the reward or minimizing the risk. The crucial feature of reinforcement learning is delayed reward or trial-and-error search. Reinforcement learning is a behavioral learning model typically applied to teach a machine to finish a multiple steps process where there are clearly predefined rules. A well-known example of reinforcement learning is computers beating humans on computer games in a super-human state. The recommendation on YouTube is an important implementation. After you watching a video on the platform, it will show you some similar titles which you are more likely to interest in. On the other hand, if you start watching the recommendation video and leave it unfinished. The machine will considering that you are not glad to watching the recommendation and another approach will be tried instead.

## 3.2 Natural Language Processing

The study of natural language processing has been around for more than 50 years and there are mainly three stages symbolic NLP, Statistical NLP and Neural NLP. Neural NLP is the most recently one. NLP is a techniques for dealing with data to figure out its hidden meaning. It aims to handle all grammatical and semantic structure, sentiment analysis etc of linguistic concepts and simulates the human ability to create natural language text. Because of this it can be separated into two parts : Natural Language Understanding and Natural Language Generation. Natural language processing can be used to summarize information or participate in a dialogue. Over the past ten years, it has been more and more professional with products such as Siri, Cortana and Google's voice search which applies NLP to understand and respond to user requests. There are billions of text data being generated every hours in social media (Facebook, YouTube etc.), blogs, news publishing platforms, google searches, messages apps (Skype, WeChat, Telegram etc.), forums (Quora, Reddit etc.) and many other ways. With the large size of text data as well as the highly unstructured data source, it is quite hard for researches to understand them by using the common and simple approach. But NLP could help to deal with such kind of data. Natural Language Processing will continue to play a significant role in the future.

The following table shows us some of the most frequently researched tasks with a rough classification in natural language processing. Some of them have immediate real-world applications, while others are more usually used as sub-tasks to help solve the larger tasks. Although the natural language processing tasks are closely intertwined, but for convenience, they can be subdivided into several categories.

**Table 1** Common research tasks in Natural Language Processing

<b>Category of Tasks</b>	<b>Tasks</b>	<b>Task description</b>
Text and speech processing	Optical character recognition	It is the task to determine the corresponding text with an given image representing printed text.
	Text-to-speech	Text-to-speech can be implied to help the visually impaired when transforming those units and producing a spoken representation with the given text.
Morphological analysis	Lemmatization	Lemmatization is a technique for reducing words to corresponding normalized form. It is the task of removing inflectional endings only and to return the base dictionary form of a work known as a lemma.
	Part-of-speech tagging	Many words can serve as multiple parts of speech. Especially the common ones, for example, "sit" can be a noun ("May I have a sit?") or verb ("sit down") and "off" can be any of at least four different parts of speech. This task is to determine the part of speech (POS) for each word in the given sentence.
Syntactic analysis	Grammar induction	The task is to generate a formal grammar which describes the syntax of the language.
	Parsing	With a given sentence, the task is to determine the parse tree (grammatical analysis) of it. For natural languages, normally the grammar is ambiguous and there are several different potential analyses for sentences. Even some sentences may have thousand of possible parses. Two primary types of parsing are grouped : constituency parsing and dependency parsing. The former one centers on building the parse tree by implying a probabilistic context-free syntax. The later one centers on the relationships between words in a sentence.
Lexical semantics (of individual words in context)	Lexical semantics	The task is to identify what is the computational meaning of individual words in context.
	Word sense disambiguation	Most of the words have multiple meanings; the meaning which is the most meaningful in context is selected. For this problem, usually we get a list of words with associated meanings, such as from a dictionary or an online resource.



Relational semantics (semantics of individual sentences)	Relationship extraction	Given a large amount of text, the task is to identify the relationships among named entities (e.g. who is the boss of whom).
Discourse (semantics beyond individual sentences)	Coreference resolution	The task is to decide which words refer to the same objects with the given sentence or huge amount of text. The general task of co-reference resolution involves identifying so-called bridging relationships involving reference expressions. Take the sentence such as "He entered Peter's house through the back window " as example, " the back window " is a referring expression and the bridging relationship to be identified is the fact that the window being referred to is the back window of John's house (rather than of some other structure that might also be referred to).
	Discourse analysis	The topics can be splitted into several related tasks. A task is the discourse analysis which identify the associated text discourse structure, for example, the nature of the discourse relations between sentences (e.g. elaboration, explanation, contrast) . Another task is the identification and classification of speech behavior in a text (for example, yes - no problem, content question, presentation, etc.)
Higher-level NLP applications	Automatic summarization (text summarization)	The task is to generate a precise and brief summary of a set of text. It is implied to produce summaries of the text with the known type, such as scientific papers, research articles.

### 3.3 Text Summarization(Automatic summarization)

As we can see in the table above, text summarization also named as automatic summarization is a common research task categorized in higher-level NLP application. Nowadays, there are a tremendous volume of data gathering in the digital space, it is very urgent to implement machine learning algorithms that can automatically summarize longer texts into a shorter summary and deliver precise summaries that can fluently express the key messages. Text summarization is right here for solving this problem. It is a technique to deal with the large amount of source text to get a short summary for better understanding. Its goal is to generate a concise, fluent and not redundant summary with only the vital information outlined in the document. People usually have to waste large amount time reading, searching and checking whether the large documents contain the information that they are looking for because of the enormous amount of textual materials. For example, you want to quickly reading a quit long news but you do not have plenty of free time and also you do not want to miss something important inside the news, text summarization does help to save your time by reduce much of this news text to shorter, focused summaries that capture the bullet details.

Text summarization methods can be divided into two groups: Extractive and Abstractive methods. Extractive text summarization method is a traditional method developed earlier. The main purpose is to identify the sentences which is the most coherent to the meaning of the source text by ranking the relevance of sentences and then choose them to generate the summary. The summary obtained includes extracted sentences from the original text. Comparing with extractive text summarization, Abstractive text summarization method is more challenging and advanced. This approach is to identify the important sections, interpret the context and reproduce a new brief text which conveys the most critical information in the original text. It guarantees that the core information is expressed through shortest text possible. The sentences in summary are newly generated instead of the extracted ones from original text.

Text summarization is truly important in our daily life. Applying text summarization is helpful to reduce reading time, accelerate the process of researching for information, and increase the amount of information.

The basic steps to perform extraction-based text summarization:

- 1) Convert the paragraph into sentences  
With a given paragraph, splitting it into its corresponding sentences normally by the punctuation.
- 2) Text processing  
For the text processing step, the stop words especially the common words with insignificant meaning such as 'a' and 'or' should be removed. It aims to filtering redundant and unimportant message which is not valuable to the text's meaning.
- 3) Tokenization  
In this step, the sentences is processed to get all the words present in the sentences.
- 4) Evaluate the weighted occurrence frequency of the words  
With the list of words processed in the previous step, the weighted scores of all the words are calculated by dividing the occurrence frequency of each of the words by the frequency of the most recurrent word in the paragraph.
- 5) Substitute words with their weighted frequencies  
Each of the words which appears in the original sentences of the paragraph is substitute with its corresponding weighted importance score. And then the sum will be computed. In the processing stage, the stop words and special characters are removed. So in this step, the weighted importance scores of the unimportant words is not considered any more.
- 6) Find sentence carries the most weight in the document based on the sum of the weighted frequencies of the words  
The summary is the collection of the sentences which are selected base on the sum value of the weighted importance scores of the words. Finally, the paragraph is reduced to an extractive summary.

# 4. Proposed Method

## 4.1 Overview

In our study, we take the paper text as input. In the step of pre-processing, the paper in TXT format is splitted into sentences, next unsupervised text summarization method is implemented to assign an importance score for each sentences in the academic paper, and then we generate the presentation slides using the Integer Linear Processing(ILP) model with several pre-defined constraints. At the same time, we also use different supervised learning models for the calculation of weighted scores of each sentences. Our work is aimed to compare the performance of our unsupervised summarization algorithm with supervised learning algorithms, to identify whether an unsupervised summarization algorithm could outperform a supervised algorithm and finally to find system with better performance for automatically extracting presentation slides from scientific papers.

The following two pictures will show the architecture of our systems with either unsupervised or supervised learning models.

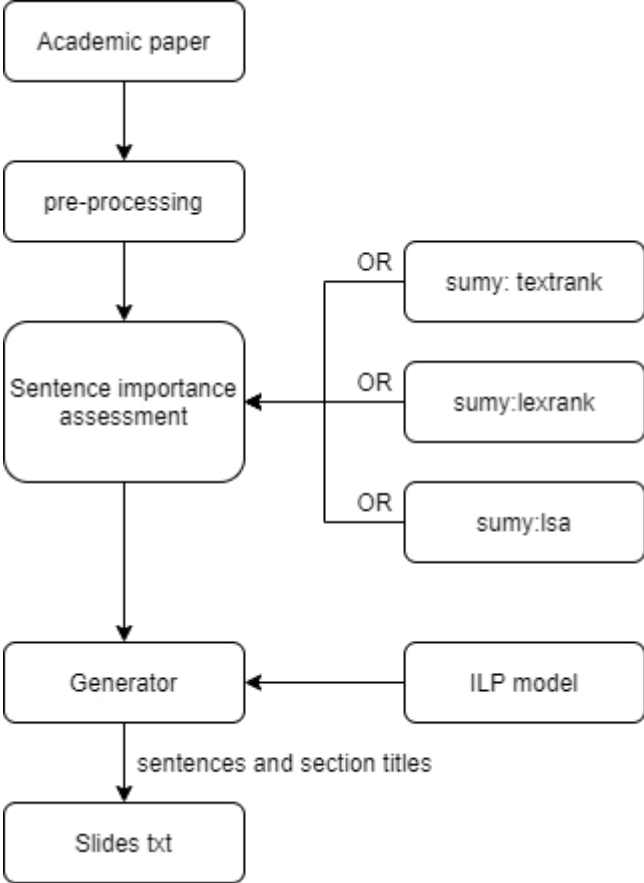
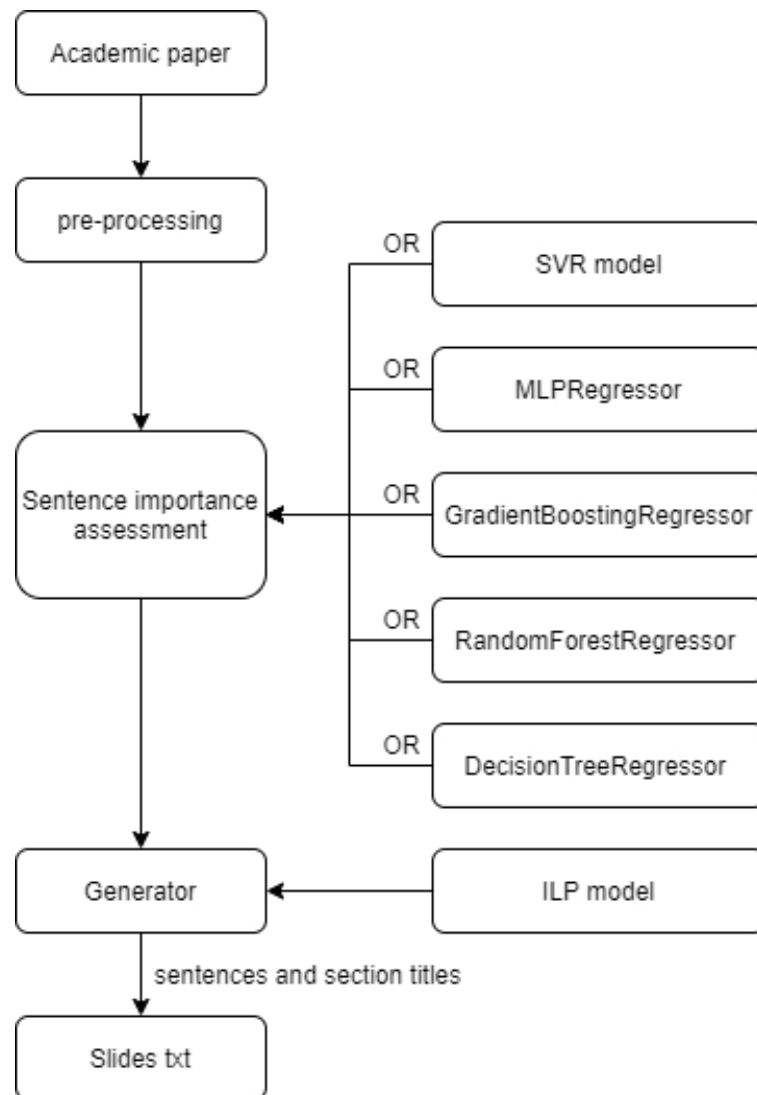


Figure 2 System architecture with unsupervised learning models



**Figure 3** System architecture with supervised learning models

## 4.2 Datasets

In order to carry on the research of automatically generating presentation slides for scientific papers. We built up a dataset that is a collection of scientific paper-slides pairs. Researches are more tend to share their study results such as scientific papers and the corresponding presentation slides in their personal pages. The personal pages' URLs are collected from an online platform Arnetminer<sup>1</sup>. With the collected homepages, we download the papers and the associated slides to build the dataset either manually or using several patterns to extract the links. There are also some other open resource such as SlideSpawn<sup>2</sup>, 52 Paper<sup>3</sup> and Sciknow 2019<sup>4</sup> where we collect the paper-slides pairs.

<sup>1</sup> <https://www.aminer.org/>

<sup>2</sup> <https://github.com/hairav/SlideSpawn>

<sup>3</sup> <https://github.com/52paper/52paper.github.io>

<sup>4</sup> <https://sciknow.github.io/sciknow2019/>

## 4.3 Datasets pre-processing

Dataset preprocessing is mainly aimed to parse the text content in paper and slides documents. The collected papers are in PDF format while the slides are either the TXT format which are crawled from website directly, PPT or PDF format. For slides in PPT or PDF format, we have to process the text content. These slides in PPT format are converted to PDF files firstly.

In order to extract the textual message as accurate as possible from the paper documents. Many tools including pdftotext<sup>5</sup>, Pdfminer<sup>6</sup>, Grobid<sup>7</sup> are tested for processing PDF file into TXT file. Finally the better performance ones are selected. With the papers in PDF format, Grobid has an evident advantage for parsing the text from a paper. So all the papers in PDF format are firstly converted into XML format with the help of Grobid [Patrice Lopez et al., 2009], and then parsing the papers in XML format into sections, for example abstract, introduction, related work, conclusion and so on, these sections are in TXT format. All the titles including paper title, sections titles and subsections titles are parsed into a text file. Sections titles are saved in another single text file which could be helpful in future section splitting work. While Grobid is not suitable for parsing the text of converted presentation PDF file. So all the slides in PDF format are processed into TXT format by using the python library named PDFMiner which is a text extraction tool for PDF documents. Finally, we got totally 195 paper-slides pairs.

Basic steps for datasets pre-processing :

- Convert papers in PDF format into XML format

A machine learning library named Grobid(GeneRation Of Bibliographic Data) is used for converting PDF into structured XML/TEI encoded documents. This library is mainly for extracting, parsing and re-structuring raw documents. It is especially efficient for dealing with technical and scientific publications.

- Extract the textual content from XML format papers

In the previous step, Grobid have the functionality for the full text extraction and structuring from PDF articles. It includes a model for the whole document segmentation and models for the structuring of the text body (paragraph, section titles, reference callout, figure, table, etc.). A python library named Beautiful Soup<sup>8</sup> is implemented for dealing with paper in XML format, the full text of paper is parsed with several patterns. We also parse all the titles which including the paper title, section titles and subsection titles with specified patterns. The section splitting can be done based on the section titles. In the end, with a given paper in XML format, we will get a text document which contains all three formats of titles, a text document which contains only section titles, a text document which contains the full paper text and several text document which contains the splitted sections' text of the paper.

- Convert slides in PDF format into TXT format

Grobid is efficient for processing technical and scientific publications but it does not work well on converting slides in PDF format. PDFMiner is a text extraction tool for PDF documents developed in python language. It is easy to installed and use. So PDFMiner is implemented for parsing the text of slides in this step.

---

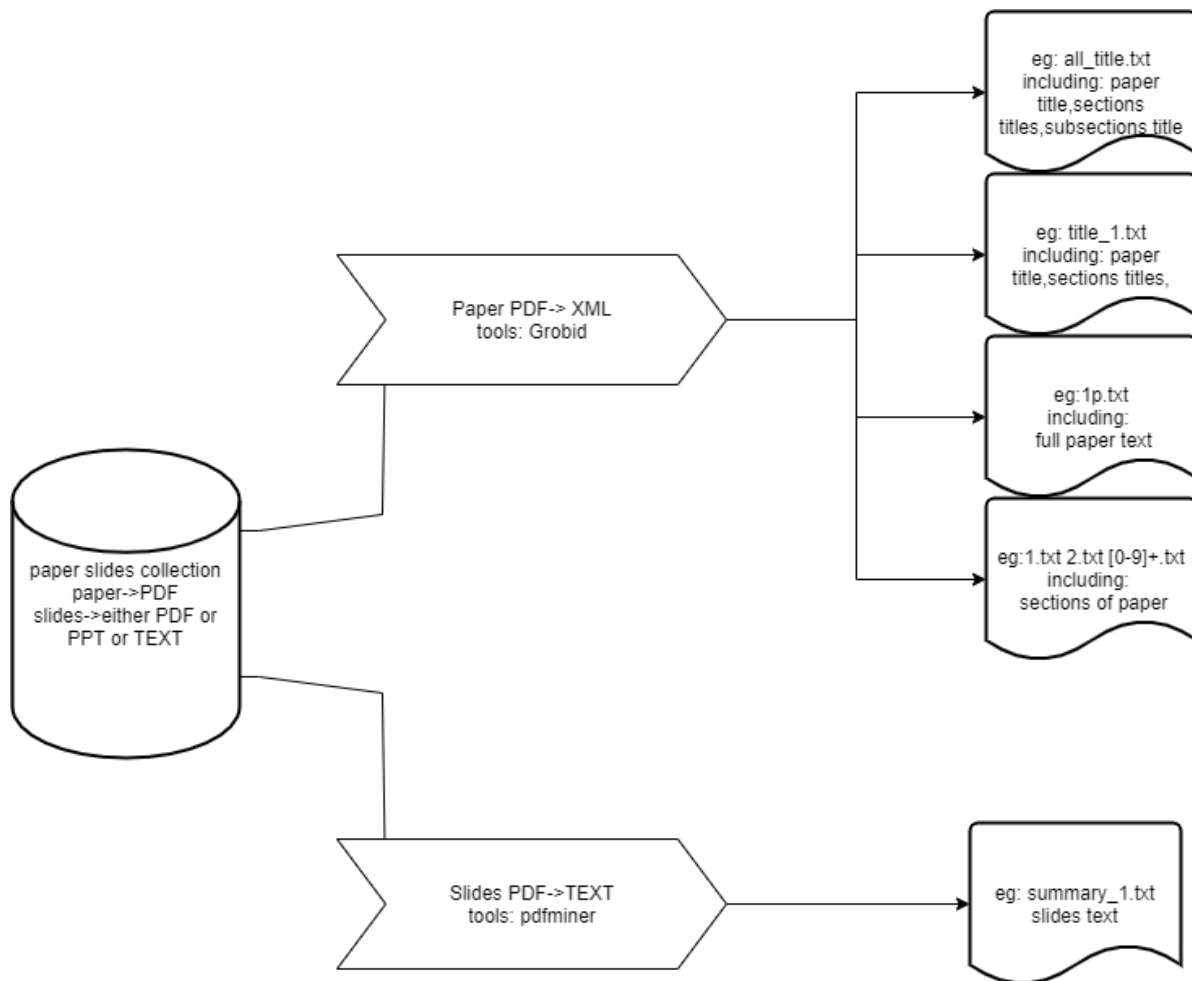
<sup>5</sup> <https://pypi.org/project/pdftotext/>

<sup>6</sup> <https://pypi.org/project/pdfminer/>

<sup>7</sup> <https://grobid.readthedocs.io/en/latest/>

<sup>8</sup> <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

The dataset pre-processing details are showed below.



**Figure 4** Datasets pre-processing

## 4.4 Sentence importance assessment

Assigning an importance score to the sentences in the paper is one of the two vital steps in our proposed system. The importance score will be utilized to generate presentation slides. Sentences with higher importance score are more likely to be selected. In our work, we have implemented different methods to compute the importance score for each sentences. The methods can be classified into two groups: unsupervised learning models and supervised learning models.

### 4.4.1 Unsupervised learning models

In our experiment, three unsupervised learning models Lsa, LexRank, TextRank are implemented. Each models contains some valuable parameters, our tests are done by changing related parameters in the reasonable range. With all these tests, the model with the best setting can be identified.

Sumy<sup>9</sup> is a text summarization library which has multiple algorithms built in. It can be used to extract summary for HTML pages or plain texts. Our unsupervised learning models takes Sumy libraries as reference.

### **Latent Semantic Analysis, LSA**

LSA [S. T. Dumais, 2004] is an Algebraic method, it is advanced but complicated. It extracts the hidden semantic structures of words and sentences.

The algorithm of LSA includes the following three major steps.

#### 1) Creation of the input matrix

The first step is to create matrix of the paper text, in our experiment each row of the matrix represents a unique word and each column stands for a sentence. Each cell contains the number of occurrence of the corresponding unique words of its row appears in the sentence identified by its column. Next, the TF metrics is computed for each sentence in the given matrix.

#### 2) Singular value decomposition(SVD)

LSA applies singular value decomposition(SVD) to the matrix. This is a form of factor analysis, or more properly the mathematical generalization of which factor analysis is a special case.

#### 3) Sentence selection

With the results of SVD, the ranks of sentences which will be considered as sentence importance matrix can be computed. And then, the ILP model is applied to the sentence selection procedure. The weighted matrix computed is used in ILP model for finally getting the most important sentences.

### **LexRank**

LexRank [Erkan et al., 2004] is an unsupervised graph-based approach for automatic text summarization. The scoring of sentences is calculated using the graph method. LexRank is used for computing sentence importance based on the concept of eigenvector centrality in a graph representation of sentences.

The algorithm used to calculate LexRank sentence importance scores are described in the following figure.

---

<sup>9</sup> <https://github.com/miso-belica/sumy>

```

1. MInput An array S of n sentences, cosine threshold t output : An array of LexRank scores;
2. Array CosineMatrix[n][n]; //the required array for Cosine matrix
3. Array Degree[n]; //the array for Degree
4. Array L[n]; //the array of Eigen values
5. /*initialize the matrix by tf-idf modified values for the given sentences if the values are greater
6. than a threshold required than the value is replaced by 0 thus create a typical tf-idf table or
7. matrix in these steps*/
8. for i← 1 to n do
9.   for j← 1 to n do
10.    CosineMatrix[i][j] = idf-modified-cosine(S[i],S[j])
11.    If CosineMatrix[i][j] > t then
12.      CosineMatrix[i][j] = 1
13.      Degree[i]++;
14.    end
15.  else
16.    CosineMatrix[i][j] = 0;
17.  end
18. end
19. End
20. /*from 17 to 21 each value of cosine matrix is divided by degree of each node*/
21. /*degree centrality here is the corresponding degree of each node*/
22. for i← 1 to n do
23.   for j← 1 to n do
24.    CosineMatrix[i][j] = CosineMatrix[i][j]/Degree[i]
25.  end
26. end
27. /*finally calculate the final score through the power iteration method*/
28. L = PowerMethod(CosineMatrix,n,q);
29. return L;

```

**Figure 5** LexRank algorithm

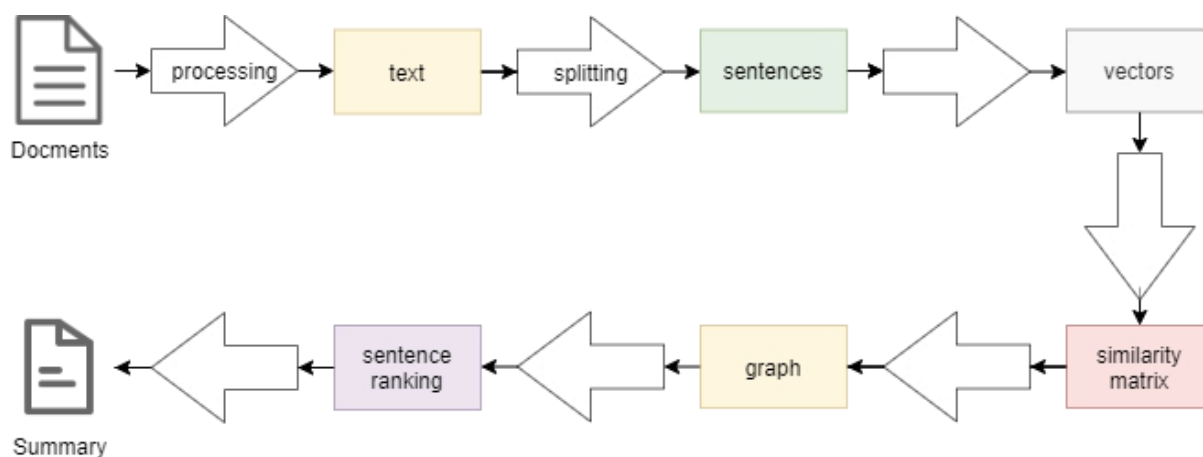
## TextRank

TextRank [Mihalcea,R. et al., 2004] is introduced as a graph-based ranking algorithm which is essentially a way of evaluating the importance of a vertex within a graph, it is based on global information recursively drawn from the entire graph.

TextRank is an unsupervised keyword extraction algorithm. It identifies how similar each sentence is to all other sentences in the text. The sentence which is the most similar to all the others will be considered as the most important sentence. The similarity function should be oriented to the semantic of the sentence, cosine similarity based on a bag of words approach can work well.

PageRank algorithm which is used to compute the rank of web pages is the foundation of TextRank algorithm. Here are the fundamentals of PageRank :The main idea is that the important pages are linked by important pages(recurrent definition). The PageRank value of a page is essentially the probability of a user visiting that page. From PageRank to TextRank, the sentences are considered as the web pages and the probability of going from sentence  $S_i$  to sentence  $S_j$  is equal to the similarity of two sentences. Finally PageRank algorithm is applied over this sentence graph.





**Figure 6** The flow of the TextRank algorithm

TextRank algorithm can be described as follows:

- 1) The first step would be to concatenate all text contained in the document.
- 2) Next, the text should be split into individual sentences .
- 3) And then, find vector representation for each sentence,
- 4) After that the similarities between sentence vectors should be calculated and stored in a similarity matrix
- 5) similarity matrix is then converted into a graph,with sentences as vertices and similarity scores as edges
- 6) The graph is used for sentence rank calculation. Finally, a certain number of top-ranked sentences will form the final summary.

#### 4.4.2 Supervised learning models

The supervised learning models used in tests are provided by scikit-learn [Pedregosa, F. et al., 2011] library<sup>10</sup>, a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. There are various supervised learning models provided. We are mainly focus on SVR, MLP, Decision Tree, Random Forest and Gradient Boosting.

##### Data training and model learning

The training data construction is based on the paper-slides pairs, the sentence scoring method is applied to assign the importance score to all the sentences in the paper. The sentences in paper which are the most similar to the sentences in slides will be assign a higher score.

The following are the features which are considered for each sentence:

- 1) Title similarity :

The paper title, section titles and subsection titles are considered three types of titles. We use the section and subsection titles which includes the sentence. The cosine similarity values between the sentence and three types of tiles are used as different features. The cosine similarity score is calculated with stop words removed and all the words stemmed.

- 2) Common with titles :

The number of words shared by the sentence and all the types of titles

<sup>10</sup> [https://scikit-learn.org/stable/supervised\\_learning.html#supervised-learning](https://scikit-learn.org/stable/supervised_learning.html#supervised-learning)

3) Total number of words after removing stop words

It is the total number of words without the stop words.

4) The Length of sentence

5) The percentage of stop words

The percentage of stop words in the total word list of the sentence.

6) Sentence position

With the position of sentence in the paper and the number of sentences in paper, the sentence position is computed as the position of sentence in the paper divided by the number of sentences in paper.

7) Number of noun phrases and verb phrases

It is the number of noun phrases and verb phrases identified by Textblob library<sup>11</sup>.

The supervised learning models will be learned based on the above mentioned features and the importance scores of sentences in the training data, and then the model is implemented to predict an importance score for each sentence in any other paper in the test datasets. This score will indicate whether the sentence will be chose for making slides or not .

## 4.5 Slides Generation

After asserting an importance score for each sentences in the paper, Integer Linear Programming(ILP) is used to select the most important sentences and keywords for generating the presentation slides. In our work, the paper full text will be separated into sections according to the section titles. The sentences in each sections will be labeled by its section title. Once a sentence is selected the corresponding section title will be considered as the title of slides.

We are applying Textblob library to the sentences and noun phrases which appears more than once are extracted as the candidate key phrases. Two kinds of phrases are defined: global phrases and local phrases. A global phrase in a specified section is recognized as a local phrase. A unique phrase (e.g. “ILP”) has a global phrase identifier (“ILP”), when a unique phrase appears in different sections are considered as different local phrases (e.g. “ILP\_introduction”, “ILP\_conclusion”). The most important and more possible to be selected global phrase is the one that linked to more local phrases. The local phrases is considered as key points for corresponding sections and the global phrases is used to address the importance differences between different unique phrases.

The proposed ILP Model are represented as follows, there are object function and constraints.

$$\max_{lp,x} \lambda_1 \sum_{i=1}^n \frac{l_i}{L_{\max}} w_i x_i + \lambda_2 \sum_{i=1}^{|B|} \frac{c_{b_i} b_i}{|B^*|} + \lambda_3 \sum_{i=1}^n \frac{w_i}{n} y_i \quad (1)$$

Subject to:

---

<sup>11</sup> <https://textblob.readthedocs.io/>

$$\sum_{i=1}^n l_i x_i \leq L_{\max} \quad (2)$$

$$\sum_{lp_j \in LP_i} lp_j \geq x_i, \text{ for } i = 1, \dots, n \quad (3)$$

$$\sum_{s_i \in s_j} x_i \geq lp_j, \text{ for } j = 1, \dots, |LP| \quad (4)$$

$$\sum_{lp_j \in LP_k} lp_j \geq y_k, \text{ for } k = 1, \dots, n \quad (5)$$

$$\sum_{b_m \in B_i} b_m \geq |B_i| x_i, \text{ for } i = 1, \dots, n \quad (6)$$

$$\sum_{s_i \in s_m} x_i \geq b_m, \text{ for } m = 1, \dots, |B| \quad (7)$$

$$\sum_{lp_j \in gp_t} lp_j \geq gp_t, \text{ for } t = 1, \dots, |GP| \quad (8)$$

$$gp_t \geq lp_j, \text{ for } \forall lp_j \in gp_t; t = 1, \dots, |GP| \quad (9)$$

$$\sum_{i=1}^{|GP|} gp_i * 2 \leq \sum_{i=1}^n x_i \quad (10)$$

$$x_i, lp_j, y_k, b_m, gp_t \in \{0, 1\}, \forall i, j, k, m, t$$

Where:

$w_i$	It represents the importance score of sentence $S_i$ which is learned by the supervised learning models such as MLP model
$n$	It represents the total number of sentences contains in the given paper
$l_i$	It is the length of sentence $S_i$
$x_i$	It tells whether sentence $S_i$ is selected for generating slides, 1 means selected otherwise not selected
$lp_j$	It is the variable that shows whether global phrase is included in slides
$y_k$	It represents the variable that identified whether sentence contains at least one selected local phrase
$b_m$	The variable that indicates whether bigram is included in the slides or not
$L_{\max}$	the maximum word count of the system generated slides
$c_{b_i}$	The number of the appearances of bigram in the paper
$B^*$	The total set of bigrams in the paper
$B$	The set of unique bigrams with duplicated bigrams removing in
$LP$	The list of all the local phrases in the paper
$GP$	The list of all the global phrases in the paper

The object function (1) can be grouped into three parts. Each part has the following special meanings.

- The first part represents the sum of the weighted scores of the selected sentences. It aims to maximize the overall importance score of the generated slides. The sentence length is considered as a multiplication factor for penalizing the very short sentences instead of simply calculating the sum of the scores.
- The second part maximizes the total weights of the bigrams which appear both in the paper and the slides. The intuition is that when more bigrams are present in the slides, the sentences in the slides are less redundant.  $c_{b_i}$  can be regarded as the weight of the bigram  $b_i$ . The more important bigrams is considered in the slides.
- The last part aims to maximize the weighted coverage of the selected key phrases . A sentence is covered by a phrase when this sentence contains the phrase. The slides which covers the content in the paper as much as possible is considered as High-quality slides. We describe this kind of coverage by using the sum of the scores of the sentences that contains the selected key phrases.

By using the maximum length  $L_{\max}$ , the total number of bigrams  $|B^*|$  and the number of the sentences  $n$ , respectively, all the terms in the object function are normalized to  $[0, 1]$ . The parameters for tuning the three groups are represented by the values of  $\lambda_1, \lambda_2$  and  $\lambda_3$ .

Here are the explanation of each constraints:

- Constraint (2) :  $L_{\max}$  is the largest total word count allowed in the generated slides.
- Constraint (3) :  $LP_i$  is the group of phrases that sentence  $s_i$  includes. It guarantees that at least one local phrase  $LP_i$  in is chose if sentence  $s_i$  is chose.
- Constraint (4) :  $s_j$  is the list of sentences that holds phrase  $lp_j$ . It guarantees that there will be at least one sentence in  $s_j$  is chose when local phrase  $lp_j$  is chose.
- Constraint (5) : The list of phrases that sentence  $s_k$  contains represented by  $LP_k$ . It guarantees that only if at least one local phrase in  $LP_k$  is included then  $y_k$  is set to value of 1.
- Constraints (6), (7) : The two constraints are close to constraints (4) and (5), separately.  $B_i$  is the list of bigrams contained in sentence  $s_i$ . The list of sentences contain bigram  $b_m$  is stored in  $S_m$ . When constraint (6) is satisfied, if  $s_i$  is selected then the total bigrams that  $s_i$  has are selected. Constraint (7) endures that when  $b_m$  is selected then at least one sentence in  $S_m$  is selected.
- Constraints (8), (9) : These two constraints holds that the corresponding global phrase is also selected when a local phrase is selected. At the same time, at least one corresponding local phrase is selected when a global phrase is selected.
- Constraints (10) : It endures than the half the number of the sentences selected is the maximum number of the total number of the selected global phrases could selected. Not too many too many key phrases can be extracted with this constraint. The count of global phrases is used instead of local phrases. The phrases appear in several sections are selected because they are considered more important.

The above optimization problem is solved by the IBM CPLEX optimizer<sup>12</sup>. By implementing the ILP model, the key phrases and sentences which will be included in the slides are obtained.

---

<sup>12</sup> <https://www.ibm.com/analytics/cplex-optimizer>

## 5. Experimental results

### 5.1 Experimental setting

For setting up our tests, with limited number of paper-slides pairs, we implementing Leave-one-out cross-validation<sup>13</sup>(LOOCV) , every time we test with one pair of paper-slides pair, all the other paper-slides pairs except the test one will be used for training. This is a procedure when we are testing with the supervised learning models. Our tested supervised models is listed below:

- 1) SVR<sup>14</sup> regression model: with the RBF kernel in LIBSVM
- 2) MLP<sup>15</sup>: MLPRegressor
- 3) Gradient Boosting<sup>16</sup>: GradientBoostingRegressor
- 4) Decision Tree<sup>17</sup>: DecisionTreeRegressor
- 5) Random Forest<sup>18</sup>: RandomForestRegressor

#### Leave-one-out cross-validation (LOOCV)

Cross-validation<sup>19</sup> is a resampling techniques used to evaluate machine learning models on a limited data set. It mainly uses a limited data set in order to estimate how accurately the model is expected to perform in general when used to make predictions on data not used during the training of the model. In a prediction problem, A dataset of known data which is known as training dataset is given to a model for training on. And a dataset of unknown data which is called validation dataset or testing set is given to the model for testing on. Cross-validation can be grouped into two types: exhaustive and non-exhaustive cross-validation.

Our used LOOCV is distinguish into exhaustive cross-validation methods. It is a special case of leave-p-out cross-validation with  $p = 1$ . The original test data is divided into a training and a validation set for learning and testing on. Comparing with LpO cross-validation, LOO cross-validation needs less computation time but there is still a large amount of computation time needed.

Pseudo-Code-Algorithm:

**Input:**  $x$ , {vector of length  $N$  with  $x$ -values of incoming points}

$y$ , {vector of length  $N$  with  $y$ -values of the expected result}

$\text{interpolate}(x\_in, y\_in, x\_out)$ , { returns the estimation for point  $x\_out$  after the model is trained with  $x\_in$ - $y\_in$  pairs }

**Output:** Err, {estimate for the prediction error}

---

<sup>13</sup> [https://en.wikipedia.org/wiki/Cross-validation\\_\(statistics\)#Leave-one-out\\_cross-validation](https://en.wikipedia.org/wiki/Cross-validation_(statistics)#Leave-one-out_cross-validation)

<sup>14</sup> <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html>

<sup>15</sup> [https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPRegressor.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html)

<sup>16</sup> <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html>

<sup>17</sup> <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html>

<sup>18</sup> <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>

<sup>19</sup> [https://en.wikipedia.org/wiki/Cross-validation\\_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))

The supervised learning models which are trained on the training data and applied on the test data in this system with its corresponding parameters are introduced below.

**Table 2** Parameter and value of tested supervised learning models

Model	Parameter	Value
SVR	kernel	rbf
MLP	hidden_layer_sizes	(50,) (100,) (200,) (50,50) (100,100,) (200,200,)
GBR	n_estimators	10, 50, 100, 200
DTR	max_depth	None, 3, 5, 10, 100
RFR	n_estimators	10, 50, 100, 200

When we do the tests with unsupervised learning models, we have tried with different value of the meaningful parameters which could affect the performance of the algorithm in Lexrank, Textrank and LSA.

**Table 3** Parameter and value of tested unsupervised learning models

Model	Parameter	Value
LSA	smooth	varying between 0 and 0.9 with steps of 0.1
TextRank	epsilon	varying between 1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3
	damping	varying between 0 and 1 with steps of 0.1
LexRank	threshold	varying between 0 and 1 with steps of 0.1
	epsilon	varying between 0.1 and 1 with steps of 0.1

## 5.2 Evaluation metrics

ROUGE <sup>20</sup>[Chin-Yew Lin, 2004], Recall-Oriented Understudy for Gisting Evaluation is an automatic evaluation method based upon n-gram comparison. It can be used for evaluating automatic produced summary of texts or machine translation by comparing it against a set of reference summaries or a single summary. In this paper, In our evaluation procedure, pythonrouge<sup>21</sup> is used to evaluate the quality of the system generated slides. It is a Python wrapper for using ROUGE, summarization evaluation toolkit. Various types of ROUGE metrics can be evaluated in this implementation. When using the general ROUGE package, it is necessary to make an XML file. But with pythonrouge, the system summaries can be evaluated with reference summaries directly. Also ROUGE scores can be evaluated in a standard way when system summaries and reference summaries are in specific directories. The author-written slides of paper is regarded as the reference summary. We computed the Precision, Recall, F-Measure score of ROUGE-1, ROUGE-2, ROUGE-SU4 and ROUGE-L.

ROUGE-1 refers to overlap of unigrams between the system summary and reference summary. ROUGE-2 evaluates a system generated summary by matching its bigrams against the reference summary. ROUGE-SU4 matches unigrams and skip-bigrams of a generated text against reference texts. ROUGE-L measures longest matching sequence of words using LCS which does not require consecutive matches but in-sequence matches that reflect sentence

<sup>20</sup> [https://en.wikipedia.org/wiki/ROUGE\\_\(metric\)](https://en.wikipedia.org/wiki/ROUGE_(metric))

<sup>21</sup> <https://github.com/tagucci/pythonrouge>

level word order. All stop words are removed and stemming is utilized when evaluating the system generate slides contents.

In the following Figure 8, there is an example for computing the precision and recall using the overlap. With the given system summary1, system summary2 and reference summary: If we consider just the individual words, the number of overlapping words between the system summary and reference summary. By implementing the Recall computation formula, the Recall score for System Summary1 is equal to 1. It means that all the words in the reference summary have been included by the system summary. The result seems very good for a text summarization system. But it can not delivery the other side of the story. A machine system summary can be quite long with all words in the reference summary captured. While, in the system summary there are many useless words which making the summary unnecessarily verbose. And then precision plays an important role.

As for precision, how much of the system summary was in fact relevant or needed is what you are essentially measuring is. Precision is measured as the following precision computation formula. The precision score Precision1 indicates that 6 out of the 7 words in the system summary were actually relevant or needed. While The precision score Precision2 does not look so good because there are quite a few unneeded words in the System Summary2. The precision aspect becomes really crucial when you are trying to generate summaries that are concise. Therefore, it is always suggested to compute both the precision and recall and then report the F-Measure. So in our work, Precision, Recall, F-Measure score are all considered.

**System Summary1 (what the machine produced):**

the cat was found under the bed

**System Summary2 (what the machine produced):**

the tiny little cat was found under the big funny bed

**Reference Summary (gold standard — usually by humans):**

the cat was under the bed

**Recall computation formula**

number\_of\_overlapping\_words/total\_words\_in\_reference\_summary

Recall = 6/6 =1.0

**Precision computation formula**

number\_of\_overlapping\_words/total\_words\_in\_system\_summary

Precision1= 6/7 =0.86

Precision2= 6/11 =0.55

**Figure 7** Simple example of computing Recall and Precision score

### 5.3 Results and Discussion

The comparison results over F-Measure score of ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4 are presented in Table 4, Table 5, Table 6, Table 7 respectively.

The Table 4 shows that even the highest ROUGE-1 F-measure score of unsupervised learning models are smaller than the lowest ROUGE-1 F-measure score of supervised learning models. So supervised learning models outperform unsupervised learning models with obvious advantages. And RFR performs best and has little difference with both GBR and DTR. It means that the slides created by using RFR model are much more similar to the human-generated slides. TextRank has a better performance than the other two unsupervised approaches.

Considering the ROUGE-2 F-measure scores showed in Table 5. All the supervised methods perform better than unsupervised methods. RFR still performs best among all other models. LexRank is the best unsupervised approach with slightly advantages than TextRank.

The Table 6 indicates the ROUGE-L F-measure scores. RFR is the best supervised approach. TextRank is the best unsupervised approach. Unsupervised approaches are not outperforming than supervised approach.

The Table 7 the ROUGE-L F-measure scores confirms that the supervised methods perform better than unsupervised methods. RFR is the best supervised approach. Lexrank is the winner among unsupervised approaches. LSA still performs worst among all the implemented methods.

To concluded, the implemented supervised methods always performed significantly better than unsupervised ones. Among supervised methods, ensemble methods (Gradient Boosting and Random Forest and Decision Tree) performed better for all the Rouge scores. MLP and SVR performed slightly worse, probably due to model overfitting issues. As for unsupervised methods, TextRank and LexRank could be the best performance unsupervised methods with specific parameter settings and they were slightly better than each other. And LSA was always the worst performance methods for all the Rouge scores. RFR was always superior to all the other supervised approaches and had clearly advantages than unsupervised methods.

**Table 4** The ROUGE-1 F-measure scores obtained

	Model	Parameter And Value	ROUGE-1-F
<b>The Highest ROUGE-1 F-measure scores of all methods</b>	RFR	n_estimators = 50	0.23484959
	GBR	n_estimators = 100	0.234603077
	DTR	max_depth = 10	0.234469333
	MLP	hidden_layer_sizes = (100,100,)	0.23096841
	SVR	kernel = 'rbf'	0.227175179
	TextRank	epsilon = 5.00E-03 damping = 0.95	0.197023333
	LexRank	epsilon = 0.6 threshold = 0.4	0.196701692
	Lsa	SMOOTH = 0.4	0.179148974
<b>The Lowest ROUGE-1 F-measure scores of supervised methods</b>	RFR	n_estimators = 10	0.232074205
	GBR	n_estimators = 10	0.226132769
	DTR	max_depth = 100	0.211952308
	MLP	hidden_layer_sizes = (50,)	0.224516513
	SVR	kernel = 'rbf'	0.227175179



**Table 5** The ROUGE-2 F-measure scores obtained

	Model	Parameter And Value	ROUGE-2-F
<b>The Highest ROUGE-2 F-measure scores of all methods</b>	RFR	n_estimators = 100	0.128503179
	GBR	n_estimators = 100	0.126427077
	DTR	max_depth = 10	0.125069538
	MLP	hidden_layer_sizes = (100,100,)	0.122638
	SVR	kernel = 'rbf'	0.114832
	TextRank	epsilon = 1.00E-04 damping = 0.8	0.079655487
	LexRank	epsilon = 0.5 threshold = 0.3	0.079845795
	Lsa	SMOOTH = 0.2	0.061243333
<b>The Lowest ROUGE-2 F-measure scores of supervised methods</b>	RFR	n_estimators = 200	0.125715385
	GBR	n_estimators = 10	0.119026769
	DTR	max_depth = 100	0.102982821
	MLP	hidden_layer_sizes = (50,)	0.117498256
	SVR	kernel = 'rbf'	0.114832

**Table 6** The ROUGE-L F-measure scores obtained

	Model	Parameter And Value	ROUGE-L-F
<b>The Highest ROUGE-2 F-measure scores of all methods</b>	RFR	n_estimators = 50	0.226066718
	GBR	n_estimators = 100	0.225482359
	DTR	max_depth = 10	0.225074615
	MLP	hidden_layer_sizes = (100,100,)	0.222832564
	SVR	kernel = 'rbf'	0.217516872
	TextRank	epsilon = 1.00E-04 damping = 0.7	0.184508564
	LexRank	epsilon = 1 threshold = 1	0.184155282
	Lsa	SMOOTH = 0.4	0.171020718
<b>The Lowest ROUGE-2 F-measure scores of supervised methods</b>	RFR	n_estimators = 10	0.223638667
	GBR	n_estimators = 10	0.216907897
	DTR	max_depth = 100	0.204492974
	MLP	hidden_layer_sizes = (50,)	0.214389897
	SVR	kernel = 'rbf'	0.217516872

**Table 7** The ROUGE-SU4 F-measure scores obtained

	Model	Parameter And Value	ROUGE-SU4-F
<b>The Highest ROUGE-2 F-measure scores of all methods</b>	RFR	n_estimators = 100	0.11259241
	GBR	n_estimators = 100	0.111300103
	DTR	max_depth = 10	0.110779128
	MLP	hidden_layer_sizes = (100,100,)	0.108710308
	SVR	kernel = 'rbf'	0.103393897
	TextRank	epsilon = 1.00E-05 damping = 0.9	0.076757385
	LexRank	epsilon = 1 threshold = 1	0.077071179
	Lsa	SMOOTH = 0.2	0.062833231
<b>The Lowest ROUGE-2 F-measure scores of supervised methods</b>	RFR	n_estimators = 200	0.110941436
	GBR	n_estimators = 10	0.104785077
	DTR	max_depth = 100	0.093249231
	MLP	hidden_layer_sizes = (50,)	0.103348103
	SVR	kernel = 'rbf'	0.103393897

The following tables shows the Precision, Recall, F-Measure score of ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4 of all the tested models with corresponding parameters.

**Table 8** Rouge-1 results of Lsa

<b>SMOOTH</b>	<b>ROUGE-1-R</b>	<b>ROUGE-1-P</b>	<b>ROUGE-1-F</b>
<b>0</b>	0.1570479	0.19823472	0.17366646
<b>0.1</b>	0.15955097	0.19959969	0.17581528
<b>0.2</b>	0.1596961	0.19797846	0.1754338
<b>0.3</b>	0.15683513	0.19704959	0.17336287
<b>0.4</b>	0.16119503	0.20536539	0.17914897
<b>0.5</b>	0.15673739	0.2000259	0.1742921
<b>0.6</b>	0.15972595	0.20350262	0.17755467
<b>0.7</b>	0.15867656	0.20144272	0.17605118
<b>0.8</b>	0.15784805	0.20161774	0.17565246
<b>0.9</b>	0.1593361	0.20415636	0.17756487

**Table 9** Rouge-2 results of Lsa

<b>SMOOTH</b>	<b>ROUGE-2-R</b>	<b>ROUGE-2-P</b>	<b>ROUGE-2-F</b>
<b>0</b>	0.0540021	0.06739159	0.05943374
<b>0.1</b>	0.055738	0.06859062	0.0610021
<b>0.2</b>	0.05572672	0.06906467	0.06124333
<b>0.3</b>	0.04915944	0.06211672	0.05446795
<b>0.4</b>	0.05240518	0.06695359	0.05833169
<b>0.5</b>	0.05068718	0.06426369	0.05621041
<b>0.6</b>	0.05231436	0.06633133	0.05803472
<b>0.7</b>	0.05373	0.06784431	0.05948359
<b>0.8</b>	0.05262374	0.06710431	0.05854236
<b>0.9</b>	0.05422851	0.06908021	0.06028564

**Table 10** Rouge-L results of Lsa

<b>SMOOTH</b>	<b>ROUGE-L-R</b>	<b>ROUGE-L-P</b>	<b>ROUGE-L-F</b>
<b>0</b>	0.14773056	0.18624144	0.16324749
<b>0.1</b>	0.14920415	0.18649123	0.16433451
<b>0.2</b>	0.14936518	0.18553287	0.16424569
<b>0.3</b>	0.14810462	0.186542	0.16392482
<b>0.4</b>	0.15377139	0.19619072	0.17102072
<b>0.5</b>	0.14996087	0.19156292	0.1668461
<b>0.6</b>	0.15225605	0.19431108	0.16938569
<b>0.7</b>	0.15207251	0.19277856	0.16861697
<b>0.8</b>	0.15082441	0.19277333	0.167896
<b>0.9</b>	0.15223656	0.19500718	0.16962097

**Table 11** Rouge-SU4 results of Lsa

<b>SMOOTH</b>	<b>ROUGE-SU4-R</b>	<b>ROUGE-SU4-P</b>	<b>ROUGE-SU4-F</b>
<b>0</b>	0.0551861	0.070508	0.06127687
<b>0.1</b>	0.05663056	0.0713638	0.06254774
<b>0.2</b>	0.0569561	0.07141621	0.06283323
<b>0.3</b>	0.05259687	0.06722574	0.05851415
<b>0.4</b>	0.05543795	0.07205169	0.0620941
<b>0.5</b>	0.05333205	0.06907154	0.05963251
<b>0.6</b>	0.05524641	0.07142774	0.06175077
<b>0.7</b>	0.05506903	0.07091062	0.06142918
<b>0.8</b>	0.05482656	0.07104626	0.061348
<b>0.9</b>	0.05583631	0.07261472	0.0625798

**Table 12** Rouge-1 results of Lexrank

<b>Epsilon</b>	<b>Threshold</b>	<b>ROUGE-1-R</b>	<b>ROUGE-1-P</b>	<b>ROUGE-1-F</b>
<b>0.1</b>	<b>0.1</b>	0.171591744	0.222177436	0.192211282
	<b>0</b>	0.171492923	0.223044256	0.192458256

	<b>0.2</b>	0.173150462	0.223521744	0.193703487
	<b>0.3</b>	0.174526615	0.227946205	0.196166564
	<b>0.4</b>	0.171773897	0.223995795	0.193030821
	<b>0.5</b>	0.173243436	0.225730154	0.194583949
	<b>0.6</b>	0.171250308	0.222383897	0.192060769
	<b>0.7</b>	0.17222041	0.224178359	0.193337744
	<b>0.8</b>	0.172604923	0.225552821	0.194137333
	<b>0.9</b>	0.173057436	0.225003077	0.194188821
	<b>1</b>	0.172644923	0.225579692	0.194167333
<b>0.2</b>	<b>0.1</b>	0.172472974	0.223403179	0.193224974
	<b>0</b>	0.173380256	0.225922974	0.194717333
	<b>0.2</b>	0.172103538	0.222133949	0.192502051
	<b>0.3</b>	0.173559179	0.22581241	0.194742923
	<b>0.4</b>	0.173434923	0.226171692	0.194880564
	<b>0.5</b>	0.173052769	0.224517231	0.19402041
	<b>0.6</b>	0.173714205	0.225726718	0.194857795
	<b>0.7</b>	0.171987026	0.224095744	0.193156615
	<b>0.8</b>	0.17217559	0.224683282	0.193525744
	<b>0.9</b>	0.173959487	0.227126256	0.195531026
	<b>1</b>	0.170086154	0.222884769	0.191483795
<b>0.3</b>	<b>0.1</b>	0.170231641	0.220347641	0.190649744
	<b>0</b>	0.172073795	0.225086718	0.19354759
	<b>0.2</b>	0.169252308	0.218839128	0.189455949
	<b>0.3</b>	0.173085179	0.224951846	0.194139692
	<b>0.4</b>	0.172115692	0.224429333	0.193379795
	<b>0.5</b>	0.173495385	0.225976256	0.194845538
	<b>0.6</b>	0.170950872	0.222715282	0.191919231
	<b>0.7</b>	0.173655487	0.22631359	0.195049282
	<b>0.8</b>	0.17376841	0.226160256	0.195083436
	<b>0.9</b>	0.173052513	0.225296974	0.194284872
	<b>1</b>	0.17305441	0.225674359	0.194424359
<b>0.4</b>	<b>0.1</b>	0.171937846	0.222724872	0.192635026
	<b>0</b>	0.169913692	0.221098154	0.190734564
	<b>0.2</b>	0.17022959	0.220391436	0.190664
	<b>0.3</b>	0.173623487	0.225761333	0.194839128
	<b>0.4</b>	0.171199692	0.223432256	0.192446462
	<b>0.5</b>	0.171976872	0.223893282	0.19311359
	<b>0.6</b>	0.173222564	0.225587949	0.194498103
	<b>0.7</b>	0.171194513	0.223222821	0.192317231
	<b>0.8</b>	0.172700359	0.225273179	0.194057436
	<b>0.9</b>	0.173633949	0.226463641	0.195103436
	<b>1</b>	0.172461795	0.223975949	0.193461179
<b>0.5</b>	<b>0.1</b>	0.171663282	0.222029846	0.192180308
	<b>0</b>	0.171878615	0.223345385	0.192801538
	<b>0.2</b>	0.171589744	0.221969692	0.192108769
	<b>0.3</b>	0.175022513	0.228578103	0.196688205
	<b>0.4</b>	0.173206256	0.226276154	0.194778308
	<b>0.5</b>	0.171824359	0.223844308	0.192985897
	<b>0.6</b>	0.170221795	0.222398	0.191369897
	<b>0.7</b>	0.171668462	0.223180513	0.192605949
	<b>0.8</b>	0.173241795	0.225909026	0.194641282
	<b>0.9</b>	0.172642564	0.224322	0.19369759
	<b>1</b>	0.169867436	0.221731333	0.190977846
<b>0.6</b>	<b>0.1</b>	0.172467333	0.223161949	0.193149949
	<b>0</b>	0.171905795	0.224212051	0.193170974
	<b>0.2</b>	0.171401692	0.221494974	0.191812564
	<b>0.3</b>	0.172343846	0.225019077	0.193671897
	<b>0.4</b>	0.174937282	0.22844641	0.196701692
	<b>0.5</b>	0.171052872	0.22319041	0.192198051
	<b>0.6</b>	0.172029744	0.223578923	0.192992

	<b>0.7</b>	0.173129385	0.225404974	0.194406615
	<b>0.8</b>	0.173004513	0.224923026	0.194107949
	<b>0.9</b>	0.171594667	0.223607385	0.192717282
	<b>1</b>	0.169826974	0.221638923	0.190875385
<b>0.7</b>	<b>0.1</b>	0.171741795	0.222509077	0.192400974
	<b>0</b>	0.171912615	0.223822974	0.192946359
	<b>0.2</b>	0.169798	0.218824462	0.189812154
	<b>0.3</b>	0.173481744	0.226037333	0.19477841
	<b>0.4</b>	0.171826564	0.224032769	0.193064
	<b>0.5</b>	0.173454103	0.225703333	0.194706205
	<b>0.6</b>	0.172759026	0.224756359	0.193863128
	<b>0.7</b>	0.174461026	0.227364974	0.195965744
	<b>0.8</b>	0.172867949	0.226296359	0.194497744
	<b>0.9</b>	0.172904564	0.225096462	0.194137795
	<b>1</b>	0.170328154	0.221796564	0.191211744
<b>0.8</b>	<b>0.1</b>	0.170551333	0.220785641	0.191016718
	<b>0</b>	0.172965692	0.225788974	0.194419231
	<b>0.2</b>	0.171714821	0.221684667	0.192075282
	<b>0.3</b>	0.173754821	0.226560103	0.195170769
	<b>0.4</b>	0.171706769	0.224452359	0.193098923
	<b>0.5</b>	0.172555897	0.224421128	0.193700769
	<b>0.6</b>	0.171854718	0.224309897	0.193114462
	<b>0.7</b>	0.172827231	0.224488051	0.193864051
	<b>0.8</b>	0.172693333	0.225245641	0.194058513
	<b>0.9</b>	0.173726513	0.226748256	0.195263282
	<b>1</b>	0.172999897	0.225861641	0.194483846
<b>0.9</b>	<b>0.1</b>	0.171610308	0.222297897	0.192230718
	<b>0</b>	0.171920615	0.224261538	0.193168821
	<b>0.2</b>	0.17038841	0.220430564	0.190764821
	<b>0.3</b>	0.172957282	0.225227692	0.194144462
	<b>0.4</b>	0.171673436	0.223741231	0.192859538
	<b>0.5</b>	0.170210667	0.22145441	0.190995179
	<b>0.6</b>	0.172516769	0.224486564	0.193659949
	<b>0.7</b>	0.171882462	0.223496923	0.192850615
	<b>0.8</b>	0.173463949	0.226532821	0.195030359
	<b>0.9</b>	0.171602821	0.224291692	0.192972154
	<b>1</b>	0.172825949	0.224427641	0.193731846
<b>1</b>	<b>0.1</b>	0.173005692	0.225555538	0.194413333
	<b>0</b>	0.170928	0.222439436	0.191800615
	<b>0.2</b>	0.172482205	0.225087795	0.193849538
	<b>0.3</b>	0.173755846	0.22692241	0.195374923
	<b>0.4</b>	0.172466308	0.224680564	0.193716821
	<b>0.5</b>	0.174948103	0.227252359	0.196273436
	<b>0.6</b>	0.173536923	0.226181026	0.194863897
	<b>0.7</b>	0.173517897	0.226030564	0.194893897
	<b>0.8</b>	0.174741385	0.226912205	0.19599041
	<b>0.9</b>	0.173627897	0.227091282	0.195376308
	<b>1</b>	0.174493538	0.227756359	0.196109077

**Table 13** Rouge-2 results of Lexrank

<b>Epsilon</b>	<b>Threshold</b>	<b>ROUGE-2-R</b>	<b>ROUGE-2-P</b>	<b>ROUGE-2-F</b>
<b>0.1</b>	<b>0.1</b>	0.069656564	0.088879385	0.077481128
	<b>0</b>	0.06905041	0.088438769	0.076933744
	<b>0.2</b>	0.070402308	0.089617538	0.078245128
	<b>0.3</b>	0.07118441	0.091282103	0.079357026
	<b>0.4</b>	0.07041441	0.090325026	0.07854
	<b>0.5</b>	0.068854667	0.087829949	0.076616308
	<b>0.6</b>	0.069453231	0.088940718	0.077396974
	<b>0.7</b>	0.069717897	0.08930041	0.077692

	<b>0.8</b>	0.069035282	0.08888441	0.077102513
	<b>0.9</b>	0.071193538	0.091094359	0.079315179
	<b>1</b>	0.069346205	0.088792974	0.077287538
<b>0.2</b>	<b>0.1</b>	0.070097077	0.089460718	0.077988359
	<b>0</b>	0.070313077	0.090170051	0.078386
	<b>0.2</b>	0.069773487	0.08868759	0.077492564
	<b>0.3</b>	0.070821744	0.090514359	0.078833077
	<b>0.4</b>	0.071357385	0.091399385	0.079539692
	<b>0.5</b>	0.069300923	0.088436923	0.077107077
	<b>0.6</b>	0.068640513	0.087691949	0.076402667
	<b>0.7</b>	0.068236308	0.087371436	0.076024821
	<b>0.8</b>	0.069883641	0.089648154	0.077936513
	<b>0.9</b>	0.070314256	0.090240462	0.078426821
	<b>1</b>	0.069109949	0.089002359	0.077189282
<b>0.3</b>	<b>0.1</b>	0.068591744	0.087373077	0.076237846
	<b>0</b>	0.069183897	0.089058205	0.077236
	<b>0.2</b>	0.068022154	0.086499949	0.075555436
	<b>0.3</b>	0.070314308	0.089809641	0.078252154
	<b>0.4</b>	0.070862769	0.091050513	0.07908359
	<b>0.5</b>	0.068886256	0.087861744	0.076643795
	<b>0.6</b>	0.068858154	0.088229026	0.076722103
	<b>0.7</b>	0.069894051	0.089557179	0.077891128
	<b>0.8</b>	0.070233795	0.090012513	0.078298256
	<b>0.9</b>	0.069778872	0.089418	0.077773077
	<b>1</b>	0.068533436	0.087804821	0.076366923
<b>0.4</b>	<b>0.1</b>	0.069365487	0.088470872	0.077150308
	<b>0</b>	0.068998205	0.088543744	0.076940256
	<b>0.2</b>	0.068982103	0.087916718	0.076693231
	<b>0.3</b>	0.070601949	0.090035487	0.078551077
	<b>0.4</b>	0.070431128	0.090372051	0.078572718
	<b>0.5</b>	0.068761846	0.087756513	0.076526051
	<b>0.6</b>	0.070393333	0.090146513	0.078443846
	<b>0.7</b>	0.067854	0.086857949	0.075584051
	<b>0.8</b>	0.069449692	0.089024051	0.077418821
	<b>0.9</b>	0.069617641	0.089356564	0.077644769
	<b>1</b>	0.069880051	0.089202615	0.077762154
<b>0.5</b>	<b>0.1</b>	0.07054041	0.089824564	0.078399846
	<b>0</b>	0.069634923	0.089513641	0.077697692
	<b>0.2</b>	0.069424051	0.088483436	0.07718959
	<b>0.3</b>	0.071686051	0.091750256	0.079845795
	<b>0.4</b>	0.070624923	0.090751333	0.078825385
	<b>0.5</b>	0.069621026	0.089255128	0.077625436
	<b>0.6</b>	0.06821759	0.087558308	0.076067641
	<b>0.7</b>	0.06902	0.088205436	0.076840154
	<b>0.8</b>	0.070297128	0.090122821	0.078372103
	<b>0.9</b>	0.070612821	0.090424256	0.078697026
	<b>1</b>	0.068927538	0.088730513	0.076980923
<b>0.6</b>	<b>0.1</b>	0.069663179	0.088832923	0.077479128
	<b>0</b>	0.069883231	0.089653282	0.077932872
	<b>0.2</b>	0.069602513	0.088608769	0.077352154
	<b>0.3</b>	0.070367795	0.090172615	0.078412
	<b>0.4</b>	0.070844051	0.090870103	0.079025385
	<b>0.5</b>	0.067477333	0.086466513	0.075188051
	<b>0.6</b>	0.068148615	0.086815487	0.075765795
	<b>0.7</b>	0.070741077	0.090792256	0.07890959
	<b>0.8</b>	0.069495128	0.088714974	0.077328462
	<b>0.9</b>	0.068809333	0.087964	0.076618103
	<b>1</b>	0.068751795	0.087888308	0.076575487
<b>0.7</b>	<b>0.1</b>	0.069027487	0.087928667	0.076719026
	<b>0</b>	0.069161282	0.088970821	0.077190564

	<b>0.2</b>	0.068525744	0.08698841	0.076060359
	<b>0.3</b>	0.07037041	0.090023077	0.078360051
	<b>0.4</b>	0.070206154	0.089923128	0.078261077
	<b>0.5</b>	0.069134154	0.088036154	0.076856154
	<b>0.6</b>	0.069938718	0.089514769	0.077913949
	<b>0.7</b>	0.070746	0.090739487	0.078888615
	<b>0.8</b>	0.068887692	0.088282872	0.076768154
	<b>0.9</b>	0.069922872	0.089603538	0.077946872
	<b>1</b>	0.068146769	0.087059846	0.075839692
<b>0.8</b>	<b>0.1</b>	0.068778	0.087625231	0.076449436
	<b>0</b>	0.070642308	0.090974974	0.078900103
	<b>0.2</b>	0.069701436	0.088772051	0.077466718
	<b>0.3</b>	0.069673641	0.088999333	0.077540154
	<b>0.4</b>	0.069262821	0.088844564	0.077232974
	<b>0.5</b>	0.068847026	0.087906513	0.076634051
	<b>0.6</b>	0.069915846	0.089743538	0.077977487
	<b>0.7</b>	0.070532256	0.09042641	0.078638513
	<b>0.8</b>	0.069912	0.089475487	0.077887231
	<b>0.9</b>	0.070472359	0.09033559	0.078565692
	<b>1</b>	0.070243538	0.090153333	0.078358308
<b>0.9</b>	<b>0.1</b>	0.068876923	0.087774103	0.076560615
	<b>0</b>	0.069776513	0.089806051	0.077897179
	<b>0.2</b>	0.069343026	0.08840359	0.077106205
	<b>0.3</b>	0.070634154	0.090313897	0.078637641
	<b>0.4</b>	0.069209641	0.088616256	0.077126872
	<b>0.5</b>	0.067894667	0.086774821	0.075564615
	<b>0.6</b>	0.069621128	0.089109128	0.077570051
	<b>0.7</b>	0.069803538	0.089319436	0.077747692
	<b>0.8</b>	0.069035436	0.088183179	0.076856667
	<b>0.9</b>	0.067716821	0.086627795	0.075423077
	<b>1</b>	0.068784564	0.087344359	0.076344718
<b>1</b>	<b>0.1</b>	0.070341179	0.090130564	0.078426564
	<b>0</b>	0.068375026	0.087257795	0.076044821
	<b>0.2</b>	0.06986	0.089762205	0.077964821
	<b>0.3</b>	0.068515128	0.087591385	0.076295128
	<b>0.4</b>	0.070223487	0.090024974	0.078303897
	<b>0.5</b>	0.071095128	0.09087559	0.079187385
	<b>0.6</b>	0.069239846	0.088632564	0.077123128
	<b>0.7</b>	0.069405128	0.089010872	0.077399231
	<b>0.8</b>	0.070287436	0.089725436	0.078223385
	<b>0.9</b>	0.06950641	0.089198513	0.077543846
	<b>1</b>	0.071204154	0.091491795	0.079463333

**Table 14** Rouge-L results of Lexrank

<b>Epsilon</b>	<b>Threshold</b>	<b>ROUGE-L-R</b>	<b>ROUGE-L-P</b>	<b>ROUGE-L-F</b>
<b>0.1</b>	<b>0.1</b>	0.16053159	0.207896821	0.179809333
	<b>0</b>	0.160818769	0.20917359	0.180459487
	<b>0.2</b>	0.162677333	0.209918154	0.181931795
	<b>0.3</b>	0.163408308	0.213364872	0.183626718
	<b>0.4</b>	0.161019231	0.210057077	0.180950872
	<b>0.5</b>	0.162066974	0.211161538	0.182006564
	<b>0.6</b>	0.160342359	0.208369282	0.179856051
	<b>0.7</b>	0.16116441	0.209887846	0.180938154
	<b>0.8</b>	0.161559744	0.211316	0.181761179
	<b>0.9</b>	0.162105744	0.210838667	0.181902
<b>0.2</b>	<b>1</b>	0.162507128	0.212395538	0.182760103
	<b>0.1</b>	0.161687692	0.209462462	0.181124051
	<b>0</b>	0.162602974	0.211917949	0.182604821
	<b>0.2</b>	0.161530205	0.208418974	0.180631692

	<b>0.3</b>	0.161765282	0.210384564	0.181451436
	<b>0.4</b>	0.162913231	0.212669282	0.183114821
	<b>0.5</b>	0.161417641	0.209409179	0.180940821
	<b>0.6</b>	0.162759641	0.211548718	0.182563385
	<b>0.7</b>	0.160902513	0.209750256	0.180720103
	<b>0.8</b>	0.161670821	0.211000923	0.181703436
	<b>0.9</b>	0.162649795	0.212421897	0.182816872
	<b>1</b>	0.16001041	0.209681128	0.180116923
<b>0.3</b>	<b>0.1</b>	0.159543846	0.20663559	0.178697333
	<b>0</b>	0.161329641	0.210933385	0.18140559
	<b>0.2</b>	0.158651641	0.205130513	0.177567846
	<b>0.3</b>	0.161789744	0.210283538	0.181451692
	<b>0.4</b>	0.161339282	0.210534821	0.181304667
	<b>0.5</b>	0.162042103	0.21110759	0.181972667
	<b>0.6</b>	0.160017846	0.208592769	0.179661077
	<b>0.7</b>	0.162643128	0.212084308	0.182698615
	<b>0.8</b>	0.163155231	0.212361282	0.183152615
	<b>0.9</b>	0.162139128	0.211189179	0.18204441
	<b>1</b>	0.162220718	0.211703231	0.182297795
<b>0.4</b>	<b>0.1</b>	0.161372103	0.209157846	0.180816769
	<b>0</b>	0.159778667	0.207866154	0.179319487
	<b>0.2</b>	0.160061641	0.207099538	0.179208974
	<b>0.3</b>	0.162077692	0.21073959	0.181856154
	<b>0.4</b>	0.160665077	0.209841385	0.180636205
	<b>0.5</b>	0.160940513	0.209470615	0.180672974
	<b>0.6</b>	0.162376051	0.211578769	0.182336205
	<b>0.7</b>	0.160009231	0.208756103	0.179767282
	<b>0.8</b>	0.161769026	0.211213436	0.181821128
	<b>0.9</b>	0.162987846	0.212665795	0.183146821
	<b>1</b>	0.161825231	0.210078615	0.18147441
<b>0.5</b>	<b>0.1</b>	0.160600769	0.207682359	0.179755846
	<b>0</b>	0.161601077	0.21002959	0.18126759
	<b>0.2</b>	0.160830872	0.207952359	0.180002462
	<b>0.3</b>	0.163478718	0.213451436	0.183675385
	<b>0.4</b>	0.16231759	0.212137179	0.182538615
	<b>0.5</b>	0.161442513	0.210364974	0.181317077
	<b>0.6</b>	0.158816103	0.207452974	0.178506667
	<b>0.7</b>	0.160369897	0.208502205	0.179905128
	<b>0.8</b>	0.162149949	0.211628462	0.18221959
	<b>0.9</b>	0.161414718	0.209837692	0.181112718
	<b>1</b>	0.160147795	0.209141077	0.180068667
<b>0.6</b>	<b>0.1</b>	0.161657333	0.209312615	0.181068718
	<b>0</b>	0.161678923	0.210828821	0.181640462
	<b>0.2</b>	0.160988103	0.207961077	0.180105436
	<b>0.3</b>	0.161235846	0.210492667	0.181160872
	<b>0.4</b>	0.163617026	0.213685949	0.183956256
	<b>0.5</b>	0.159856615	0.208632923	0.179607538
	<b>0.6</b>	0.160822154	0.209051385	0.180405897
	<b>0.7</b>	0.162138462	0.211204154	0.182078615
	<b>0.8</b>	0.162019744	0.210754667	0.181798615
	<b>0.9</b>	0.160778615	0.20964441	0.180592821
	<b>1</b>	0.160512718	0.209557333	0.180417128
<b>0.7</b>	<b>0.1</b>	0.161056872	0.208758923	0.180439282
	<b>0</b>	0.161005026	0.209900667	0.180808615
	<b>0.2</b>	0.159351231	0.205238051	0.178064308
	<b>0.3</b>	0.162352564	0.211474974	0.182241487
	<b>0.4</b>	0.160911128	0.209805897	0.18077759
	<b>0.5</b>	0.162369692	0.211347538	0.182260462
	<b>0.6</b>	0.161677949	0.210468564	0.181446769
	<b>0.7</b>	0.163491846	0.213244359	0.183685231

	<b>0.8</b>	0.161634205	0.211712615	0.181873487
	<b>0.9</b>	0.162085949	0.211139538	0.182016923
	<b>1</b>	0.159865077	0.208149487	0.179435077
<b>0.8</b>	<b>0.1</b>	0.159792	0.206913949	0.178961282
	<b>0</b>	0.162688205	0.212473128	0.182885282
	<b>0.2</b>	0.161213128	0.208169692	0.180319692
	<b>0.3</b>	0.162341333	0.211634462	0.182312308
	<b>0.4</b>	0.161344872	0.211074256	0.181479231
	<b>0.5</b>	0.161349692	0.209865897	0.18110359
	<b>0.6</b>	0.161159795	0.210521538	0.181133282
	<b>0.7</b>	0.162061179	0.210595282	0.181793487
	<b>0.8</b>	0.16188	0.211225487	0.181916103
	<b>0.9</b>	0.162272923	0.21186041	0.182387128
	<b>1</b>	0.163300769	0.213340205	0.183610205
<b>0.9</b>	<b>0.1</b>	0.160784564	0.208307179	0.180090205
	<b>0</b>	0.161267949	0.210384667	0.181185385
	<b>0.2</b>	0.160194667	0.207118923	0.179283333
	<b>0.3</b>	0.161955641	0.210925282	0.181782462
	<b>0.4</b>	0.160614718	0.209370359	0.180424256
	<b>0.5</b>	0.159301846	0.207300154	0.178742
	<b>0.6</b>	0.162119333	0.211157846	0.182039128
	<b>0.7</b>	0.160917744	0.209383744	0.180572821
	<b>0.8</b>	0.162014	0.211625026	0.182151385
	<b>0.9</b>	0.160781538	0.210198205	0.180801692
	<b>1</b>	0.161679949	0.209940154	0.181208
<b>1</b>	<b>0.1</b>	0.161942872	0.211175026	0.181971026
	<b>0</b>	0.160302923	0.208678051	0.179876205
	<b>0.2</b>	0.161971949	0.211517179	0.182062256
	<b>0.3</b>	0.162425487	0.212213333	0.182643487
	<b>0.4</b>	0.161852615	0.210810564	0.181756462
	<b>0.5</b>	0.163626154	0.212572462	0.183551231
	<b>0.6</b>	0.162197128	0.211542308	0.182150308
	<b>0.7</b>	0.162436615	0.211787949	0.182493128
	<b>0.8</b>	0.163463282	0.212289795	0.183322256
	<b>0.9</b>	0.162794769	0.213125641	0.183240205
	<b>1</b>	0.163846872	0.213964872	0.184155282

**Table 15** Rouge-SU4 results of Lexrank

<b>Epsilon</b>	<b>Threshold</b>	<b>ROUGE-SU4-R</b>	<b>ROUGE-SU4-P</b>	<b>ROUGE-SU4-F</b>
<b>0.1</b>	<b>0.1</b>	0.066818154	0.087153744	0.074988359
	<b>0</b>	0.066798718	0.087684256	0.075161487
	<b>0.2</b>	0.066838974	0.086962564	0.074936308
	<b>0.3</b>	0.068082615	0.089674718	0.076702
	<b>0.4</b>	0.067227949	0.088343333	0.075708564
	<b>0.5</b>	0.067345077	0.088336359	0.075768462
	<b>0.6</b>	0.067136718	0.087990051	0.075506718
	<b>0.7</b>	0.06694959	0.087859949	0.075324769
	<b>0.8</b>	0.066745179	0.087982821	0.075254308
	<b>0.9</b>	0.067967026	0.08904159	0.076423538
	<b>1</b>	0.067068615	0.088112103	0.075521026
<b>0.2</b>	<b>0.1</b>	0.067361795	0.088030923	0.075662872
	<b>0</b>	0.067690462	0.08899559	0.076214974
	<b>0.2</b>	0.066363333	0.086277128	0.074368974
	<b>0.3</b>	0.067927385	0.089129333	0.076399231
	<b>0.4</b>	0.067870821	0.089152564	0.076413026
	<b>0.5</b>	0.067032051	0.087573026	0.075283385
	<b>0.6</b>	0.067095179	0.087822718	0.075406513
	<b>0.7</b>	0.066426974	0.087298615	0.074781795
	<b>0.8</b>	0.067020513	0.088083949	0.075467026



	<b>0.9</b>	0.067181897	0.088386103	0.075663077
	<b>1</b>	0.066287692	0.087475641	0.074755846
<b>0.3</b>	<b>0.1</b>	0.066357385	0.086551179	0.074465026
	<b>0</b>	0.066879026	0.088291179	0.075417897
	<b>0.2</b>	0.065094	0.084697333	0.072968564
	<b>0.3</b>	0.067459846	0.088425077	0.075848205
	<b>0.4</b>	0.067423333	0.088608513	0.075919846
	<b>0.5</b>	0.067159077	0.087963436	0.075514
	<b>0.6</b>	0.066386974	0.087031846	0.07463641
	<b>0.7</b>	0.067321692	0.088375897	0.075753949
	<b>0.8</b>	0.067470051	0.088545077	0.075919949
	<b>0.9</b>	0.067370821	0.088433128	0.075805385
	<b>1</b>	0.066233897	0.086918205	0.074511077
<b>0.4</b>	<b>0.1</b>	0.066776103	0.087120359	0.074950872
	<b>0</b>	0.066295692	0.087067436	0.074616359
	<b>0.2</b>	0.065758718	0.085688974	0.073764974
	<b>0.3</b>	0.067755282	0.088788513	0.076198718
	<b>0.4</b>	0.067160462	0.088340308	0.075662974
	<b>0.5</b>	0.066528	0.087128308	0.074803282
	<b>0.6</b>	0.067597692	0.088602359	0.076022
	<b>0.7</b>	0.066008359	0.086696462	0.074290308
	<b>0.8</b>	0.066980872	0.087996718	0.075399385
	<b>0.9</b>	0.067054103	0.088079897	0.075480923
	<b>1</b>	0.066877692	0.087403385	0.075129128
<b>0.5</b>	<b>0.1</b>	0.067243436	0.087664	0.075442615
	<b>0</b>	0.066737026	0.087635128	0.075097026
	<b>0.2</b>	0.065904615	0.085798154	0.073896103
	<b>0.3</b>	0.068338615	0.089888872	0.076936462
	<b>0.4</b>	0.067576564	0.088956462	0.07615159
	<b>0.5</b>	0.067121385	0.088170821	0.075564872
	<b>0.6</b>	0.065929077	0.086614205	0.074199487
	<b>0.7</b>	0.066309231	0.086810308	0.074528308
	<b>0.8</b>	0.067434718	0.088523744	0.07589041
	<b>0.9</b>	0.067417077	0.088288821	0.075804103
	<b>1</b>	0.066254718	0.087157897	0.074644564
<b>0.6</b>	<b>0.1</b>	0.067142154	0.087564051	0.075355641
	<b>0</b>	0.066792359	0.087824615	0.07521759
	<b>0.2</b>	0.066274308	0.086223897	0.074293897
	<b>0.3</b>	0.067198154	0.088384564	0.075655128
	<b>0.4</b>	0.068073795	0.089557744	0.076700923
	<b>0.5</b>	0.065713333	0.086304872	0.07394441
	<b>0.6</b>	0.066431077	0.086835282	0.074616718
	<b>0.7</b>	0.067476564	0.088626513	0.075961692
	<b>0.8</b>	0.067011385	0.087687692	0.075301692
	<b>0.9</b>	0.066676103	0.087449077	0.074998308
	<b>1</b>	0.066472308	0.087338769	0.074843692
<b>0.7</b>	<b>0.1</b>	0.066566154	0.086821333	0.07469359
	<b>0</b>	0.066659436	0.087577538	0.075002615
	<b>0.2</b>	0.065508103	0.08500159	0.073356974
	<b>0.3</b>	0.067504923	0.088663128	0.075956103
	<b>0.4</b>	0.067069538	0.088054513	0.075500872
	<b>0.5</b>	0.067097538	0.087728667	0.075378308
	<b>0.6</b>	0.067163795	0.088043282	0.075522821
	<b>0.7</b>	0.067713897	0.089016872	0.076245744
	<b>0.8</b>	0.06723041	0.088558513	0.075746462
	<b>0.9</b>	0.067488154	0.088544769	0.075935385
	<b>1</b>	0.066371179	0.086886872	0.074584462
<b>0.8</b>	<b>0.1</b>	0.066513333	0.086783692	0.074650462
	<b>0</b>	0.067271538	0.088615179	0.075808564
	<b>0.2</b>	0.066167077	0.086018615	0.074143231

	<b>0.3</b>	0.06740241	0.088534564	0.075849897
	<b>0.4</b>	0.066802154	0.087963333	0.075271026
	<b>0.5</b>	0.067178	0.088041436	0.075560462
	<b>0.6</b>	0.067010718	0.088063385	0.075428359
	<b>0.7</b>	0.067568667	0.088573026	0.075999641
	<b>0.8</b>	0.067430513	0.088470667	0.07587241
	<b>0.9</b>	0.067536974	0.08877	0.076044205
	<b>1</b>	0.067532974	0.088764051	0.076045487
<b>0.9</b>	<b>0.1</b>	0.066665846	0.087042154	0.074832462
	<b>0</b>	0.067152718	0.088544718	0.075694718
	<b>0.2</b>	0.065760359	0.085663795	0.073753282
	<b>0.3</b>	0.067334615	0.088303333	0.075717795
	<b>0.4</b>	0.066912667	0.08789841	0.075340256
	<b>0.5</b>	0.065843949	0.086155179	0.073967487
	<b>0.6</b>	0.067226821	0.088128051	0.075615795
	<b>0.7</b>	0.066539128	0.087121128	0.074785846
	<b>0.8</b>	0.067055846	0.088076308	0.075486821
	<b>0.9</b>	0.066341949	0.087249436	0.074706051
	<b>1</b>	0.066757641	0.087127282	0.074904051
<b>1</b>	<b>0.1</b>	0.067519949	0.088725077	0.076041282
	<b>0</b>	0.065795128	0.086052205	0.07389041
	<b>0.2</b>	0.06704441	0.088191077	0.075516872
	<b>0.3</b>	0.066683436	0.087548769	0.075051641
	<b>0.4</b>	0.067363333	0.088429385	0.075819282
	<b>0.5</b>	0.067968667	0.088925897	0.076399282
	<b>0.6</b>	0.067251538	0.088308462	0.075657846
	<b>0.7</b>	0.067234308	0.088288103	0.075685641
	<b>0.8</b>	0.067507795	0.088248205	0.075838
	<b>0.9</b>	0.067397128	0.088830615	0.075996256
	<b>1</b>	0.068401436	0.090058923	0.077071179

**Table 16** Rouge-1 results of Textrank

<b>Epsilon</b>	<b>Damping</b>	<b>ROUGE-1-R</b>	<b>ROUGE-1-P</b>	<b>ROUGE-1-F</b>
<b>1.00E-04</b>	<b>0.85</b>	0.175214051	0.227298103	0.196428154
	<b>0.9</b>	0.174044359	0.226354256	0.195332718
	<b>0.8</b>	0.174956256	0.227288154	0.196252462
	<b>0.95</b>	0.172065949	0.223369487	0.192965385
	<b>0.5</b>	0.173718872	0.226781333	0.195280308
	<b>0.55</b>	0.174638667	0.227488564	0.196117282
	<b>0.6</b>	0.172490256	0.224150615	0.193508667
	<b>0.65</b>	0.173758359	0.225474923	0.194804821
	<b>0.7</b>	0.175576359	0.228001949	0.196929333
	<b>0.75</b>	0.172018103	0.222925846	0.192750821
<b>1.00E-05</b>	<b>0.85</b>	0.173282718	0.226158821	0.194779744
	<b>0.9</b>	0.174904769	0.227475949	0.196280974
	<b>0.8</b>	0.173522256	0.225992359	0.194878821
	<b>0.95</b>	0.173368974	0.225044308	0.194431385
	<b>0.5</b>	0.172832718	0.225396769	0.194201231
	<b>0.55</b>	0.173891538	0.226989949	0.195461744
	<b>0.6</b>	0.171914205	0.223896615	0.193059846
	<b>0.65</b>	0.172087128	0.224160256	0.193255282
	<b>0.7</b>	0.173135077	0.225466821	0.194417897
	<b>0.75</b>	0.171970154	0.223603846	0.192973385
<b>5.00E-05</b>	<b>0.85</b>	0.172739846	0.225226103	0.194018154
	<b>0.9</b>	0.173144359	0.225161949	0.194294308
	<b>0.8</b>	0.173343846	0.225144256	0.194449282
	<b>0.95</b>	0.174876564	0.22758041	0.196304974
	<b>0.5</b>	0.170451333	0.222043077	0.191415436
	<b>0.55</b>	0.170956205	0.222163282	0.191804359
	<b>0.6</b>	0.172891231	0.224995538	0.194070667

	<b>0.65</b>	0.174739538	0.227375538	0.196134564
	<b>0.7</b>	0.171724564	0.222765128	0.192511128
	<b>0.75</b>	0.173496308	0.225774564	0.194784974
<b>5.00E-04</b>	<b>0.85</b>	0.173637795	0.225669128	0.194773487
	<b>0.9</b>	0.173569692	0.226021385	0.194881026
	<b>0.8</b>	0.173502564	0.224834821	0.194411949
	<b>0.95</b>	0.173614923	0.225625846	0.194753641
	<b>0.5</b>	0.173525744	0.226003744	0.194823641
	<b>0.55</b>	0.172600308	0.224452667	0.193696103
	<b>0.6</b>	0.173009231	0.224994308	0.194138667
	<b>0.65</b>	0.174008718	0.226815179	0.195468308
	<b>0.7</b>	0.172749026	0.224404821	0.193767179
	<b>0.75</b>	0.172795436	0.223912821	0.193599077
<b>1.00E-03</b>	<b>0.85</b>	0.173538821	0.22572241	0.194752923
	<b>0.9</b>	0.17297041	0.223928	0.193700513
	<b>0.8</b>	0.174197641	0.226282821	0.195406718
	<b>0.95</b>	0.173555846	0.225397026	0.194672051
	<b>0.5</b>	0.172338308	0.223666308	0.193227436
	<b>0.55</b>	0.171383487	0.223002256	0.192340923
	<b>0.6</b>	0.173102923	0.22518041	0.194257282
	<b>0.65</b>	0.172963487	0.224136821	0.193792667
	<b>0.7</b>	0.173891333	0.225833231	0.194954872
	<b>0.75</b>	0.172014923	0.222950205	0.192784308
<b>5.00E-03</b>	<b>0.85</b>	0.173562513	0.225932821	0.194831795
	<b>0.9</b>	0.173892205	0.225478718	0.194911282
	<b>0.8</b>	0.174332051	0.226744769	0.195643385
	<b>0.95</b>	0.175438051	0.228604821	0.197023333
	<b>0.5</b>	0.172181744	0.223942513	0.193250308
	<b>0.55</b>	0.170780051	0.222088923	0.191648462
	<b>0.6</b>	0.173686872	0.225851179	0.194874462
	<b>0.65</b>	0.173084359	0.224197333	0.193892564
	<b>0.7</b>	0.171351436	0.222585026	0.192189487
	<b>0.75</b>	0.173671077	0.225903179	0.194886974

**Table 17** Rouge-2 results of Textrank

<b>Epsilon</b>	<b>Damping</b>	<b>ROUGE-2-R</b>	<b>ROUGE-2-P</b>	<b>ROUGE-2-F</b>
<b>1.00E-04</b>	<b>0.85</b>	0.070916513	0.090505949	0.078910872
	<b>0.9</b>	0.070426205	0.090117641	0.078440718
	<b>0.8</b>	0.071540513	0.091441385	0.079655487
	<b>0.95</b>	0.070161744	0.089751026	0.078143846
	<b>0.5</b>	0.070139385	0.090025487	0.078235077
	<b>0.55</b>	0.070388769	0.090128667	0.078424103
	<b>0.6</b>	0.069810256	0.089338205	0.077762462
	<b>0.65</b>	0.070641641	0.09004041	0.078559436
	<b>0.7</b>	0.070489538	0.089918769	0.07842041
	<b>0.75</b>	0.070460974	0.090015231	0.078431949
<b>1.00E-05</b>	<b>0.85</b>	0.069492564	0.089133949	0.077492154
	<b>0.9</b>	0.070608051	0.090261385	0.078619744
	<b>0.8</b>	0.069415077	0.088870564	0.077344462
	<b>0.95</b>	0.070182	0.089892359	0.078216872
	<b>0.5</b>	0.068845333	0.088287744	0.076758667
	<b>0.55</b>	0.069589077	0.089337231	0.077620256
	<b>0.6</b>	0.068912256	0.08819841	0.076765487
	<b>0.65</b>	0.069655231	0.089446103	0.077695795
	<b>0.7</b>	0.069055436	0.088532308	0.076971385
	<b>0.75</b>	0.068839538	0.088041231	0.076654308
<b>5.00E-05</b>	<b>0.85</b>	0.069821897	0.089439897	0.077796923
	<b>0.9</b>	0.069825897	0.08919959	0.077719333
	<b>0.8</b>	0.070378615	0.089790718	0.078305538

	<b>0.95</b>	0.070931333	0.090963641	0.079077436
	<b>0.5</b>	0.068183385	0.087381333	0.075990154
	<b>0.55</b>	0.069419487	0.088805487	0.077315282
	<b>0.6</b>	0.068354051	0.087452359	0.076122821
	<b>0.65</b>	0.070294205	0.089873692	0.078276667
	<b>0.7</b>	0.068878667	0.087832769	0.076600564
	<b>0.75</b>	0.070183487	0.089894923	0.078224667
<b>5.00E-04</b>	<b>0.85</b>	0.070587128	0.090055179	0.078514615
	<b>0.9</b>	0.070500256	0.090242769	0.078537436
	<b>0.8</b>	0.070243538	0.089656769	0.078154513
	<b>0.95</b>	0.069958769	0.08960041	0.077943026
	<b>0.5</b>	0.069664051	0.089008513	0.077540667
	<b>0.55</b>	0.069459846	0.088983641	0.077401231
	<b>0.6</b>	0.070394051	0.090071128	0.078399231
	<b>0.65</b>	0.070802974	0.090697385	0.078909692
	<b>0.7</b>	0.070693385	0.090270513	0.078677846
	<b>0.75</b>	0.070871436	0.090417846	0.078837026
<b>1.00E-03</b>	<b>0.85</b>	0.070764974	0.090491846	0.078795641
	<b>0.9</b>	0.069882	0.089140564	0.07771959
	<b>0.8</b>	0.071471179	0.091303333	0.079562718
	<b>0.95</b>	0.070514	0.090239231	0.078548051
	<b>0.5</b>	0.069514769	0.088744256	0.07734159
	<b>0.55</b>	0.069160359	0.088309795	0.076947692
	<b>0.6</b>	0.070236308	0.089872462	0.078231385
	<b>0.65</b>	0.070026615	0.08923841	0.077858154
	<b>0.7</b>	0.070509179	0.089986462	0.078443077
	<b>0.75</b>	0.070325385	0.089705487	0.078243436
<b>5.00E-03</b>	<b>0.85</b>	0.06988641	0.089139026	0.077736308
	<b>0.9</b>	0.06929841	0.088260667	0.077030103
	<b>0.8</b>	0.070862308	0.090653231	0.078926205
	<b>0.95</b>	0.071030205	0.09093	0.079133179
	<b>0.5</b>	0.069290462	0.088711282	0.077202513
	<b>0.55</b>	0.068466154	0.087579385	0.076240564
	<b>0.6</b>	0.070048	0.089551282	0.077989436
	<b>0.65</b>	0.070136051	0.089337333	0.077969231
	<b>0.7</b>	0.069728923	0.089091744	0.077614769
	<b>0.75</b>	0.070422615	0.090043282	0.078405231

**Table 18** Rouge-L results of Textrank

<b>Epsilon</b>	<b>Damping</b>	<b>ROUGE-L-R</b>	<b>ROUGE-L-P</b>	<b>ROUGE-L-F</b>
<b>1.00E-04</b>	<b>0.85</b>	0.164366821	0.213311385	0.184276974
	<b>0.9</b>	0.162277179	0.210997179	0.182077179
	<b>0.8</b>	0.164105128	0.21326759	0.184084513
	<b>0.95</b>	0.161405487	0.209533692	0.180984667
	<b>0.5</b>	0.163312154	0.213277795	0.18358759
	<b>0.55</b>	0.163992308	0.213649897	0.184144923
	<b>0.6</b>	0.161539692	0.209969077	0.181214154
	<b>0.65</b>	0.163116205	0.211564513	0.182814308
	<b>0.7</b>	0.164505487	0.213686462	0.184508564
	<b>0.75</b>	0.161169897	0.208883128	0.180576513
<b>1.00E-05</b>	<b>0.85</b>	0.162293333	0.211996872	0.182466923
	<b>0.9</b>	0.163526154	0.212755333	0.183513538
	<b>0.8</b>	0.162392462	0.211599436	0.182390513
	<b>0.95</b>	0.162554821	0.21106841	0.182302359
	<b>0.5</b>	0.161884103	0.211262615	0.181924769
	<b>0.55</b>	0.163355641	0.213334564	0.183626769
	<b>0.6</b>	0.160904872	0.209590513	0.180682359
	<b>0.65</b>	0.161536667	0.210453641	0.181399179
	<b>0.7</b>	0.162209692	0.211351385	0.182165077

	<b>0.75</b>	0.161016923	0.209473385	0.180697385
<b>5.00E-05</b>	<b>0.85</b>	0.161937846	0.211147538	0.181866308
	<b>0.9</b>	0.161941949	0.210642	0.181714308
	<b>0.8</b>	0.162838256	0.211492051	0.182635385
	<b>0.95</b>	0.163673487	0.213073231	0.18372841
	<b>0.5</b>	0.159781436	0.208134256	0.179406051
	<b>0.55</b>	0.160484513	0.208722256	0.180093897
	<b>0.6</b>	0.162186667	0.211262974	0.182103692
	<b>0.65</b>	0.163763436	0.213155077	0.183812872
	<b>0.7</b>	0.160854872	0.208747026	0.180329077
	<b>0.75</b>	0.162703538	0.211677538	0.182622923
<b>5.00E-04</b>	<b>0.85</b>	0.162369128	0.211180667	0.182167231
	<b>0.9</b>	0.161950205	0.210824872	0.181786462
	<b>0.8</b>	0.162294923	0.210330051	0.181838308
	<b>0.95</b>	0.162272718	0.210933795	0.182021026
	<b>0.5</b>	0.162969436	0.212231179	0.182939179
	<b>0.55</b>	0.161752154	0.210397487	0.181515949
	<b>0.6</b>	0.161917436	0.21066559	0.181701231
	<b>0.65</b>	0.163597897	0.213251231	0.183752051
	<b>0.7</b>	0.162585385	0.211230103	0.182352564
	<b>0.75</b>	0.161465795	0.209238872	0.180881538
<b>1.00E-03</b>	<b>0.85</b>	0.162568462	0.211432821	0.182410769
	<b>0.9</b>	0.161645795	0.209332154	0.181013641
	<b>0.8</b>	0.163444872	0.212234923	0.18329041
	<b>0.95</b>	0.161797744	0.210109897	0.181446974
	<b>0.5</b>	0.161362154	0.209497128	0.180923641
	<b>0.55</b>	0.160690667	0.209173436	0.180346103
	<b>0.6</b>	0.162172923	0.210972256	0.181973333
	<b>0.65</b>	0.162012205	0.209940821	0.181490154
	<b>0.7</b>	0.162951077	0.211800923	0.182752051
	<b>0.75</b>	0.161372821	0.209119897	0.180817128
<b>5.00E-03</b>	<b>0.85</b>	0.162735231	0.21194441	0.18269159
	<b>0.9</b>	0.162382	0.210548923	0.181979026
	<b>0.8</b>	0.163579282	0.212765231	0.183549487
	<b>0.95</b>	0.164099026	0.213843692	0.184270051
	<b>0.5</b>	0.161783897	0.210431385	0.181559641
	<b>0.55</b>	0.15994959	0.208024051	0.179478769
	<b>0.6</b>	0.162696205	0.211629795	0.182543641
	<b>0.65</b>	0.16236841	0.210397077	0.181893436
	<b>0.7</b>	0.160546462	0.208569641	0.180055179
	<b>0.75</b>	0.162423385	0.211289538	0.182242513

**Table 19** Rouge-SU4 results of Textrank

<b>Epsilon</b>	<b>Damping</b>	<b>ROUGE-SU4-R</b>	<b>ROUGE-SU4-P</b>	<b>ROUGE-SU4-F</b>
<b>1.00E-04</b>	<b>0.85</b>	0.067908308	0.088961179	0.076359538
	<b>0.9</b>	0.06750641	0.088615897	0.075963077
	<b>0.8</b>	0.068164308	0.089326923	0.076650769
	<b>0.95</b>	0.067053128	0.087913077	0.075422103
	<b>0.5</b>	0.067362923	0.088630103	0.075881897
	<b>0.55</b>	0.067956872	0.089239128	0.076482256
	<b>0.6</b>	0.066910564	0.087719692	0.075254718
	<b>0.65</b>	0.067480769	0.088257026	0.075813744
	<b>0.7</b>	0.068227231	0.089345795	0.07670759
	<b>0.75</b>	0.067052359	0.087704872	0.075340564
<b>1.00E-05</b>	<b>0.85</b>	0.067413487	0.088709949	0.075947436
	<b>0.9</b>	0.06823841	0.089492103	0.076757385
	<b>0.8</b>	0.067182615	0.088157641	0.075599231
	<b>0.95</b>	0.067638667	0.088676769	0.076089231
	<b>0.5</b>	0.067077744	0.088257128	0.075562103

	<b>0.55</b>	0.067571744	0.088990462	0.076141538
	<b>0.6</b>	0.066881641	0.087821538	0.075276513
	<b>0.65</b>	0.066995385	0.088023077	0.075418308
	<b>0.7</b>	0.067444821	0.088661795	0.075945026
	<b>0.75</b>	0.066602103	0.087261179	0.074882923
<b>5.00E-05</b>	<b>0.85</b>	0.067250051	0.088471949	0.075726256
	<b>0.9</b>	0.067062667	0.087876051	0.07540559
	<b>0.8</b>	0.067988513	0.089062	0.076456359
	<b>0.95</b>	0.068011077	0.089381897	0.076568718
	<b>0.5</b>	0.066038256	0.086725692	0.074324615
	<b>0.55</b>	0.066759744	0.087476	0.075071538
	<b>0.6</b>	0.066915538	0.087821897	0.07528959
	<b>0.65</b>	0.068122308	0.089427949	0.076656462
	<b>0.7</b>	0.066732051	0.087219436	0.074953231
	<b>0.75</b>	0.067355077	0.088389538	0.075800359
<b>5.00E-04</b>	<b>0.85</b>	0.067524	0.088533846	0.075938462
	<b>0.9</b>	0.067519333	0.088727179	0.076006718
	<b>0.8</b>	0.067308	0.087965795	0.075599795
	<b>0.95</b>	0.06744759	0.088556769	0.075894564
	<b>0.5</b>	0.067462256	0.088546205	0.075897077
	<b>0.55</b>	0.067209846	0.088245436	0.075639897
	<b>0.6</b>	0.067520564	0.088623436	0.075966513
	<b>0.65</b>	0.067543282	0.08875959	0.076042923
	<b>0.7</b>	0.067403128	0.088364103	0.075804974
	<b>0.75</b>	0.067533692	0.088272923	0.075850308
<b>1.00E-03</b>	<b>0.85</b>	0.067578821	0.088644923	0.076019026
	<b>0.9</b>	0.066967231	0.087512769	0.075200564
	<b>0.8</b>	0.067903077	0.088939744	0.076348103
	<b>0.95</b>	0.067626923	0.088673949	0.076072256
	<b>0.5</b>	0.067263949	0.088002308	0.075581795
	<b>0.55</b>	0.066777795	0.087523795	0.075081897
	<b>0.6</b>	0.067593282	0.088781385	0.076071179
	<b>0.65</b>	0.067444564	0.088208103	0.075770513
	<b>0.7</b>	0.067700974	0.088596718	0.076051692
	<b>0.75</b>	0.067125077	0.087814769	0.075440513
<b>5.00E-03</b>	<b>0.85</b>	0.067083487	0.087929231	0.075432359
	<b>0.9</b>	0.067287333	0.087978205	0.075594513
	<b>0.8</b>	0.067844154	0.08907841	0.076348462
	<b>0.95</b>	0.068088051	0.089513641	0.076656462
	<b>0.5</b>	0.067009846	0.087970872	0.075417026
	<b>0.55</b>	0.066180718	0.086785487	0.074435641
	<b>0.6</b>	0.067526359	0.088553641	0.075941949
	<b>0.65</b>	0.067155641	0.087720513	0.075404205
	<b>0.7</b>	0.066771179	0.087501385	0.075078667
	<b>0.75</b>	0.067703231	0.088815846	0.076155744

**Table 20** Rouge-1 results of supervised learning models

Model	Parameter	Value	ROUGE-1-R	ROUGE-1-P	ROUGE-1-F
SVR	kernel	rbf	0.206136103	0.256957128	0.227175179
MLP	hidden_layer_sizes	(100, )	0.209212769	0.259166	0.230040821
MLP	hidden_layer_sizes	(50,)	0.203839282	0.253461385	0.224516513
MLP	hidden_layer_sizes	(50,50,)	0.207256667	0.254572769	0.22706359
MLP	hidden_layer_sizes	(100,100,)	0.210380462	0.259833282	0.23096841
MLP	hidden_layer_sizes	(200,200,)	0.209102974	0.258545641	0.229775077
MLP	hidden_layer_sizes	(200,)	0.20668041	0.255934974	0.227132051
GBR	n_estimators	10	0.205364923	0.255668821	0.226132769
GBR	n_estimators	50	0.210804103	0.259672821	0.231113077
GBR	n_estimators	100	0.213951282	0.263594103	0.234603077
GBR	n_estimators	200	0.213114769	0.262832308	0.233822103

DTR	max_depth	NONE	0.193682462	0.239615282	0.212645641
DTR	max_depth	3	0.210927282	0.262487333	0.232388667
DTR	max_depth	5	0.212360872	0.262678256	0.233269744
DTR	max_depth	10	0.214097282	0.263183436	0.234469333
DTR	max_depth	100	0.193194256	0.239037128	0.211952308
RFR	n_estimators	10	0.21208	0.260405538	0.232074205
RFR	n_estimators	50	0.214216513	0.263922769	0.23484959
RFR	n_estimators	100	0.214329846	0.262286718	0.23423559
RFR	n_estimators	200	0.213605949	0.261805487	0.233738154

**Table 21** Rouge-2 results of supervised learning models

Model	Parameter	Value	ROUGE-2-R	ROUGE-2-P	ROUGE-2-F
SVR	kernel	rbf	0.104003487	0.130100513	0.114832
MLP	hidden_layer_sizes	(100, )	0.109471179	0.135780923	0.12049041
MLP	hidden_layer_sizes	(50,)	0.106314462	0.133143795	0.117498256
MLP	hidden_layer_sizes	(50,50,)	0.109405692	0.134978923	0.120124154
MLP	hidden_layer_sizes	(100,100,)	0.111503077	0.138230872	0.122638
MLP	hidden_layer_sizes	(200,200,)	0.110910205	0.137974051	0.122245128
MLP	hidden_layer_sizes	(200,)	0.107981641	0.134165897	0.118874256
GBR	n_estimators	10	0.107877128	0.134738462	0.119026769
GBR	n_estimators	50	0.113452872	0.140195179	0.124574667
GBR	n_estimators	100	0.114999949	0.142450205	0.126427077
GBR	n_estimators	200	0.113771949	0.140950205	0.125099385
DTR	max_depth	NONE	0.095707538	0.118159128	0.105020769
DTR	max_depth	3	0.110056256	0.137684667	0.121579538
DTR	max_depth	5	0.111458256	0.137937744	0.122471949
DTR	max_depth	10	0.114122667	0.140452513	0.125069538
DTR	max_depth	100	0.094147231	0.115621692	0.102982821
RFR	n_estimators	10	0.116790769	0.143144974	0.127726256
RFR	n_estimators	50	0.116340564	0.143372667	0.127625026
RFR	n_estimators	100	0.117381026	0.144195692	0.128503179
RFR	n_estimators	200	0.114736615	0.140945436	0.125715385

**Table 22** Rouge-L results of supervised learning models

Model	Parameter	Value	ROUGE-L-R	ROUGE-L-P	ROUGE-L-F
SVR	kernel	rbf	0.197269744	0.246153282	0.217516872
MLP	hidden_layer_sizes	(100, )	0.201246	0.249607487	0.221407641
MLP	hidden_layer_sizes	(50,)	0.194495333	0.242249744	0.214389897
MLP	hidden_layer_sizes	(50,50,)	0.198676513	0.244247897	0.217747641
MLP	hidden_layer_sizes	(100,100,)	0.202841744	0.250859744	0.222832564
MLP	hidden_layer_sizes	(200,200,)	0.200135333	0.247742256	0.220044359
MLP	hidden_layer_sizes	(200,)	0.198201385	0.245881179	0.218000154
GBR	n_estimators	10	0.196785795	0.24550759	0.216907897
GBR	n_estimators	50	0.202968154	0.250268564	0.222621897
GBR	n_estimators	100	0.205510462	0.253504154	0.225482359
GBR	n_estimators	200	0.204878667	0.252795436	0.224836462
DTR	max_depth	NONE	0.186535897	0.230936718	0.204861538
DTR	max_depth	3	0.201866718	0.251545231	0.222545026
DTR	max_depth	5	0.203281692	0.251868359	0.223465846
DTR	max_depth	10	0.205426051	0.252761538	0.225074615
DTR	max_depth	100	0.186378154	0.230654051	0.204492974
RFR	n_estimators	10	0.204336615	0.250983744	0.223638667
RFR	n_estimators	50	0.206098718	0.25418641	0.226066718
RFR	n_estimators	100	0.206725128	0.253205692	0.226013128
RFR	n_estimators	200	0.205017026	0.251621487	0.224480256

**Table 23** Rouge-SU4 results of supervised learning models

Model	Parameter	Value	ROUGE-1-R	ROUGE-1-P	ROUGE-1-F
SVR	kernel	rbf	0.093254	0.118029077	0.103393897
MLP	hidden_layer_sizes	(100, )	0.09692159	0.12174	0.107191077
MLP	hidden_layer_sizes	(50,)	0.093155179	0.117884256	0.103348103
MLP	hidden_layer_sizes	(50,50,)	0.096505026	0.120059385	0.106260256
MLP	hidden_layer_sizes	(100,100,)	0.098477282	0.123327949	0.108710308
MLP	hidden_layer_sizes	(200,200,)	0.09699159	0.121845282	0.107286615
MLP	hidden_layer_sizes	(200,)	0.094888	0.119189846	0.104874615
GBR	n_estimators	10	0.094475231	0.119696103	0.104785077
GBR	n_estimators	50	0.099595436	0.124086564	0.109653231
GBR	n_estimators	100	0.100969744	0.126098154	0.111300103
GBR	n_estimators	200	0.10022759	0.125173538	0.110498923
DTR	max_depth	NONE	0.085832462	0.107283282	0.094599128
DTR	max_depth	3	0.097598462	0.123428513	0.108238462
DTR	max_depth	5	0.098902359	0.123730974	0.109080821
DTR	max_depth	10	0.100669128	0.125350615	0.110779128
DTR	max_depth	100	0.084759436	0.105766051	0.093249231
RFR	n_estimators	10	0.102016205	0.126571692	0.112074667
RFR	n_estimators	50	0.102015538	0.127295744	0.112430103
RFR	n_estimators	100	0.102509795	0.127098769	0.11259241
RFR	n_estimators	200	0.100798821	0.125287641	0.110941436

## 6. Conclusions and future works

The objectives of this paper was to design, implement and evaluate a new automated approach to automatically generate slides of scientific paper based on unsupervised document summarization and compare its performance with that of the supervised models. Evaluation results are based on 195 paper-slides pairs to verify the performance of our proposed methods.

The F-Measure scores of ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4 confirm that the implemented supervised methods performed significantly better than unsupervised ones. Among supervised methods, the ensemble methods such as Gradient Boosting and Random Forest and Decision Tree performed better for all the Rouge scores. Probably due to model overfitting issues, MLP and SVR performed slightly worse. RFR was always superior to all the other supervised approaches and had evident advantages than unsupervised methods. As for unsupervised methods, TextRank and LexRank could be the best performance unsupervised methods and they were slightly better than each other. And LSA was always the worst performance methods for all the Rouge scores.

For the future work, we would like to make much more efforts in the following four aspects. Firstly, in our proposed system some extractive methods like LexRank, LSA, Textrank have been tested, so we can implement abstractive text summarization methods to extend current work. Secondly, it could be helpful to increase the performance of proposed system with a large paper-slides pairs collected and used for training and testing. Thirdly, in the proposed system only textual contents are focused on, because of the rapid growth of multimedia data, Multi-modal summarization has been more attractive. We could also work on adding equations, figures, charts and tables into the generated slides to make the slides more comprehensive and attractive. Lastly, we could implement some domain-specified models to improve the performance of our current system.



## Bibliography

- [1] [Yue Hu et al., 2015] Yue Hu and Xiaojun Wan. 2015. PPSGen: Learning-based presentation slides generation for academic papers. *IEEE transactions on knowledge and data engineering* 27, 4 (2015), 1085–1097. <https://www.ijcai.org/Proceedings/13/Papers/310.pdf>
- [2] [Horacio, A et al., 2012] Horacio Saggion, Thierry Poibeau. *Automatic Text Summarization: Past, Present and Future*. T. Poibeau; H. Saggion. J. Piskorski, R. Yangarber. *Multi-source, Multilingual Information Extraction and Summarization*, Springer, pp.3-13, 2012, *Theory and Applications of Natural Language Processing*, 978-3-642-28569-1. fihal-00782442
- [3] [Hotho, A et al., 2015] Hotho, A., Nürnberger, A. and Paaß, G. (2005). "A brief survey of text mining". In *Ldv Forum*, Vol. 20(1), p. 19-62
- [4] [Yasumura et al., 2003] Yasumura, Yoshiaki ; Takeichi, Masashi ; Nitta, Katsumi. / A support system for making presentation slides. In: *Transactions of the Japanese Society for Artificial Intelligence*. 2003 ; Vol. 18, No. 4. pp. 212-220.
- [5] [T. Shibata et al., 2005] T. Shibata and S. Kurohashi, "Automatic slide generation based on discourse structure analysis", *Proc. Int. Joint Conf. Natural Lang. Process.*, pp. 754-766, 2005.
- [6] [D. Galanis et al., 2012] D. Galanis, G. Lampouras and I. Androutsopoulos, "Extractive multi-document summarization with integer linear programming and support vector regression", *Proc. COLING*, pp. 911-926, 2012.
- [7] [Autade Dhanshri P et al., 2016] Autade Dhanshri P, Prof. Raut S.Y, "SLIDEGen: Approach to automatic Slides Generation in *International Research Journal of Engineering and Technology (IRJET)* Volume: 03 Issue: 01— Jan 2016.
- [8] [Athar Sefid et al., 2019] Athar Sefid, Jian Wu, Prasenjit Mitra, and C Lee Giles. 2019. Automatic slide generation for scientific papers.
- [9] [Shaj et al., 2020] Shaj, Kevin, and Sara Susan John. "Learning Based Slide Generator."
- [10] [TJ Fu et al., 2021] TJ Fu, WY Wang, D McDuff, Y Song -DOC2PPT: Automatic Presentation Slides Generation from Scientific Documents arXiv:2101.11796, 2021 - arxiv.org
- [11] [Cagliero, L., & La Quatra, M., 2020] Cagliero, L., & La Quatra, M. (2020). Extracting highlights of scientific articles: A supervised summarization approach. *Expert Systems with Applications*, 160, 113659.
- [12] [Mao, X. et al., 2019] Mao, X., Yang, H., Huang, S., Liu, Y., & Li, R. (2019). Extractive summarization using supervised and unsupervised learning. *Expert Systems with Applications*, 133, 173-181.
- [13] [Jain, A. et al., 2017] Jain, A., Bhatia, D., & Thakur, M. K. (2017, December). Extractive text summarization using word vector embedding. In *2017 International Conference on Machine Learning and Data Science (MLDS)* (pp. 51-55). IEEE.
- [14] [Collins, E. et al., 2017] Collins, E., Augenstein, I., & Riedel, S. (2017). A supervised approach to extractive summarisation of scientific papers. *arXiv preprint arXiv:1706.03946*.
- [15] [Cagliero, L. et al., 2019] Cagliero, L., Garza, P., & Baralis, E. (2019). ELSA: A multilingual document summarization algorithm based on frequent itemsets and latent semantic analysis. *ACM Transactions on Information Systems (TOIS)*, 37(2), 1-33.

- [16][Erkan, G. et al., 2004] Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22, 457-479.
- [17][Tixier, A. et al., 2017] Tixier, A., Meladianos, P., & Vazirgiannis, M. (2017, September). Combining graph degeneracy and submodularity for unsupervised extractive summarization. In *Proceedings of the workshop on new frontiers in summarization* (pp. 48-58).
- [18][Miller, D., 2019] Miller, D. (2019). Leveraging BERT for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*.
- [19][Zheng, Hao et al., 2019] Zheng, Hao, and Mirella Lapata. "Sentence centrality revisited for unsupervised summarization." *arXiv preprint arXiv:1906.03508* (2019).
- [20][F. Barrios et al., 2016] F. Barrios, F. Lopez, L. Argerich, and R. Wachenchauser, "Variations of the similarity function of textrank for automated summarization," *arXiv preprint arXiv:1602.03606*, 2016
- [21][S. Lamsiyah et al., 2020] S. Lamsiyah, A. El Mahdaouy, B. Espinasse, and S. El Alaoui Ouatik, "An unsupervised method for extractive multi-document summarization based on centroid approach and sentence embeddings," *Expert Systems with Applications*, p. 114152, 2020. [Online].
- [22][Patrice Lopez et al., 2009] Patrice Lopez. 2009. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *Proceedings of the 13th European Conference on Research and Advanced Technology for Digital Libraries, ECDL'09*, pages 473–474
- [23][Erkan et al., 2004] G. Erkan and D. R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479
- [24][Mihalcea, R. et al., 2004] Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing Order into Texts. In L. Dekang & W. Dekai (Eds.), *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)* (pp. 8-15). Association for Computational Linguistics.  
<https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf>
- [25][S. T. Dumais, 2004] S. T. Dumais, "Latent semantic analysis," *Annual review of information science and technology*, vol. 38, no. 1, pp.188–230, 2004
- [26][Chin-Yew Lin, 2004] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out* (2004).
- [27][Pedregosa, F. et al., 2011] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12:2825-2830