Streamlined Slide Generation by Performing Extractive Text Summarization

Lalithamani Nithyanandham, Aniruddha Madduri, Sainadh Makineni, Sanjay Bhargav Patibandla, M. Srinandhini

Abstract: ***Power Point Presentations have long been used by people to display information in an easy and eye-catching manner. Many organizations use the PowerPoint Presentations for discussing their projects with stakeholders, or for their project reviews. Many students use the Presentations to display their ideas of designs and how they implement it. This involves the user to analyse the content on which the presentation is to be made, shrink the content into smaller parts, and finally creating slides manually.***

There are many tools which are able to generate slides, and which automate the process of doing that. Generation of slides must involve some content, and this work aims to involve content given by the user varying from a normal text file to a web-page, and make available to the user the slides, based on the content.

Our work involves automating the process by summarizing the entire text document given as an input, and generating PowerPoint slides using the aforementioned summary.

Keywords: Extractive summary, Textmining, Text Ranking, Text Summarization.

I. INTRODUCTION

Text categorization is the analysis of finding out the respective category of the given text according to their content. The categories present is often called as controlled vocabulary. No information is used except for the content present in the document itself during the classification of the document. The categorization of data into different categories is mainly done by the tasks required. Some tasks may require a single category whereas some tasks may require multiple categories. The main objective is to assign ranks to these categories by the estimation of the relevance of these documents to the required document. Deterministic category labelling is often compared with the probabilistic category labelling. The processing of data might happen in different ways, either it may happen one by one or it may happen collectively. There might be a possibility of assigning the task to find out the different documents present in a class or the task of finding the documents present in different classes.

Revised Manuscript Received on December 30, 2018.

Lalithamani Nithyanandham, Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India.

Aniruddha Madduri, Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India.

Sainadh Makineni, Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India.

Sanjay Bhargav Patibandla, Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India.

M. Srinandhini, Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India.

Text mining is a field where meaningful information is mined from the document present. It may simply be characterized as the data analysis and extraction of the information required for the purpose to be filed. The difference between data in databases and data in documents is that, databases can only store data in a structured format, while data in documents need not be structured. In modern times, text is the most common way or the path for the exchange of information.

II. LITERATURE SURVEY

A survey of various text-summarization methods has been analysed. Here we present a survey on few of them. Vishal Gupta,GurupreetSinghLehel experimented on Extractive Summarisation methods (TF-IDF, Cluster Based method, Graph theoretic approach etc.) which selects important paragraphs and sentences from the document and it concatenates into smaller meaningful sentences. [1]

Razieh Abbasi-Ghalehtaki et al tested on mathematical models for systems which interact between each other for calculating similarity of sentences and particle swarm optimization method for weighting to the features for their importance and use fuzzy logic for sentence scoring. Text summarization based combination of fuzzy, particle swarm optimization and cellular learning automata is proved better compared to text summarization based cellular learning automata. [2]

Rasim M. Alguliyev et al analyzed document summarization as a Boolean programming problem. A modified differential evolution is created to solve the optimization problem. The properties like redundancy, length, relevance are optimized. The problem of repetition of information has been dealt. The dependency of summarization result on the similarity measure has been demonstrated. [3]

Changjian Fang et al demonstrated the co-ranking model combining the word-sentence relationship with the graph-based unsupervised ranking model. Redundancy elimination technique is presented as a supplement to co-rank. [4]

Gunes Erkhan and Dragomir R. Radev implemented a graph-based method involving lexical centrality which focusses on multi-document extractive generic text summarization. This method, called LexRank, uses cosine similarity to find the similarity between any two sentences of the input text.This similarity score is used to generate the adjacency matrix. The sentences are then selected based on the number of outgoing and incoming edges of a node (or) the degree of the node. [5]

Hongyan Jing experimented on reducing the size of sentences for text summarization. The system presented removes words, phrases from sentences which are not need, automatically. Syntactic knowledge, context information, and statistics are used to decide which phrase can be removed from the document. [6]

Lawrence H Reeve et al tested the use of text summarization to get important information from medical texts. The methods introduced in this paper *BioChain* and *FreqDist* are used to identify the theme of the sentences and to remove redundant information respectively, and the optimum output is obtained when the methods are run together [7].

Yue Hu and Xiaojun Wan employ a method involving regression to learn the scores of the sentences which are important, and integer linear programming or ILP, to obtain slides with sentences being selected and aligned properly. [8]

Ning Zhong et al worked on a method to discover patterns in the way text can be mined. Two processes help in the refinement of the discovered patterns in the text document, patter deploying and pattern evolving. This method is proposed to reduce the ineffectiveness of using the patterns (data) in the realm of text mining [9].

Ganesan et al constructed an approach to generate an abstractive summary. This involves using the words as nodes of a graph, and the sentence score is obtained by traversing through each node in the graph with respect to the unique properties of the graph, and scoring the path traversed. This is done through the entire graph, after which the duplicates sentences are taken out. The sentences are then ranked and the best scored sentences are chosen as the summary. [10]

Massih-Reza Amini and Patrick Gallinari discussed the text summarization method which extracts important sentences from the text without the need of any labelled corpora to learn the method of ranking sentences. Implementation of a similarity evaluating method and a classification model which uses self-supervised learning, allows for the extraction of the summary from the text. [11]

Priya Ganguly and Prachi M. Joshi generated PowerPoint Presentations using the text summarization algorithms to use text as an input to the slide generator. [12]

Anusha Bagalkotkar et al proposed a novel technique for efficient text document summarization. It uses the input file to get the term frequency (TF) of each term not in the list of stop-words and the sentences are extracted using the terms which appear more frequently. [13]

Mehdi Allahyari et al discussed the various methods and ways to extract summaries from a given text file. All the methods are evaluated, examined and analysed, and the success and failure of each method is discussed. [14]

Meghana Chaudary et al elaborates on the previous work of other researchers on the field of sentiment analysis and techniques of ranking. The technique used involves the ranking of users on the basis of likes and shares. Cosine similarity was one of the method of ranking used in the system. [15]

S.A.Babar et al modelled a text summarization technique which uses the fuzzy interference system. Important sentences are taken as the basis of the summary. The fuzzy logic extraction method is the main focus of the paper, along with the semantic approach to summarization of text. [16]

Chandra Sekhar Yadav et al propose a model which focusses on the single document summarization, by using an extraction based hybrid model. This model uses a mixture of term frequency, inverse document frequency, sentiment analysis and the positioning of a sentence [17].

Changjian Fang et al. proposes a novel word-sentence co-ranking model named co-rank, which combines the word-sentence relationship with the graph-based unsupervised ranking model [18].

Rasmita Rautray and Rakesh Chandra Balabantaray discussed the few meta-heuristic approaches such as Cuckoo Search (CS), Cat Swarm Optimization (CSO), Particle Swarm Optimization (PSO), Harmony Search (HS), and Differential Evolution (DE) algorithm is
presented for single document summarization problem [19].

Farshad Kiyoumarsi compared the summaries generated by automatic methods and summaries obtained with the help of many English teachers and Professors. The automatic methods involved the use of Fuzzy logic and Vector approach [20] [29].

Tian Wang et al evaluated fog based approach for trustworthy communication in sensor cloud system. They formulated the evaluation issue as a problem of linear regression. The least squares algorithm has been used to find the best trust evaluation model [21].

Rafael Ferreira and Luciano de Souza Cabral assessed sentencing scoring techniques for extractive text summarization. This paper looks into the various ways in which sentences from the input file are taken into account and scored, which is a vital step in summarizing an entire document [22].

Simone Teufel and Marc Moens experimented with the relevance and the rhetorical status of scientific articles. It presented a design which uses a training set of a list of citation. Further, this method also selects articles based on the content and groups it into seven categories [23].

Ya-Han Hua et al proposed a text summarization technique for identifying the leading-k most reviews for the hotels. Both the content and sentiment similarities were used to gauge the similarity [24].

Francesco Ronzano and Horacio Saggion researched on knowledge extraction from scientific articles. The framework supports evaluation of two processes, rhetorical sentence classification and extractive text summarization is done. [25]

M. Vamsee Krishna Kiran et al proposed a novel recommendation method of various products in the market, by considering the important terms regarding the technical features of products and classifying them based on their polarity [26].

Hans Moen et al proposed three summarization methods, each outperforming the other, to make it easier for clinicians to get an overview as well as to writing the reports for the purpose of discharge [27].

Ailin Li et al discussed the implementations of both LexRank and TextRank and proposed a method where both the methods are implemented in correlation to one another [28].

Rada Mihalcea and Paul Tarau discussed a graph based summarization technique, ranking each of the sentences using a random-surfer model, calling this method as TextRank. [30]

III. SYSTEM OVERVIEW

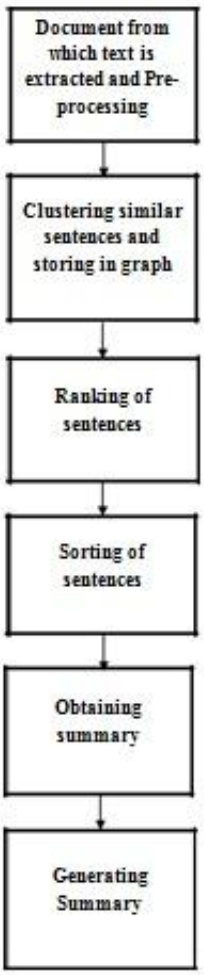

Figure 1: Structural representation of the graph based model

Fig-1 demonstrates the structural representation of the graph based approach used. This is a formal idea of how the slide generator works.

Document Input and Pre-processing: The docx file, of which the summary is to be obtained, is given as the input of the tool. The text from the docx file is taken with the help of nodes of a tree. The document.xml file of the word file is used to construct a tree containing the words (as nodes). For each node of the tree, the node is added to a list, which is used as the input for the pre-processing part. The pre-processing part includes the checking of stop words and punctuations of the list obtained. This is done to simplify the working of the model. The stop words are checked with the help of the Natural Language Toolkit (NLTK) package.

Clustering similar sentences: The text obtained previously is taken as the input, and the sentences are clustered. Clustering means to group similar sentences together. So the grouping of sentences is done by Levenshtein distance.

The idea of Levenshtein distance is to find out the minimum edit distance between any 2 nodes, in this case, sentences.

This is implemented in such a way that two nodes are being compared and the similarity score is put as the edge length, and clustering happens in such a way that the nodes with less distance between them are taken as similar nodes.

Ranking of sentences: After finding out the clusters of sentences, the summary can be obtained by ranking each cluster and taking the most important sentences in that cluster. The PageRank Algorithm, used by Google, serves as the base for this ranking to happen. PageRank works for web-pages. A derivative, TextRank is used to rank sentences. The sentences are taken in as the web-pages and then ranking takes place.

The TextRank algorithm works as follows:

a. Each sentence in the graph has an inward or an outward edge, connecting it to different nodes in the graph

b. The rank of sentence A can be found out by using the formula:

c. PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn)), where PR(A) is the rank of sentence A, ‘d’ is a damping factor, PR(T1) is the rank of sentence T1 which link to A, C(T1) is the number of outbound links to sentence T1.

d. This formula is used for all the sentences in the graph.

Sorting sentences: After the rank of all the sentences are obtained, the sentences are then sorted on the basis of their scores. Sorting is done to get the top ranked sentences up front, and this forms the basis of the summary.

Obtaining the summary: The ranking and sorting of sentences allows for the sentences which are important to come up in the first few sentences of the vector. So, when the number of sentences are given, the top ‘n’ (number of sentences) sentences will be chosen as the summary of the Word file.

Generation of slides: While the summary is created, each sentence in the top n sentences will be pushed into the PowerPoint Presentation, and for each sentence, a new slide shall be generated. After all the sentences are put in the summary and the presentation, the presentation is saved and the summary is shown.

IV. RESULTS AND DISCUSSIONS

The following section discusses the various modules used for the working of the system and the subsequent results obtained:

```
try:
    from xml.etree.cElementTree import
XML except ImportError:
    from xml.etree.ElementTree import XML
import zipfile
"""
Module that extract text from MS XML Word
document (.docx).
"""
WORD_NAMESPACE =
'{http://schemas.openxmlformats.org/wordprocessingml/2006
/main}'
PARA = WORD_NAMESPACE + 'p'
TEXT = WORD_NAMESPACE + 't'
def get_docx_text(path):
    document = zipfile.ZipFile(path)
    xml_content = document.read('word/document.xml')
    document.close()
    tree = XML(xml_content)
    paragraphs = []
    for paragraph in tree.getiterator(PARA):
        texts = [node.text
                 for node in paragraph.getiterator(TEXT)
                 if node.text]
if texts:
paragraphs.append(''.join(texts))
return '\n\n'.join(paragraphs)
```

Figure 2: Text extraction from .docx file

```
def levenshtein_distance(first, second):

    if len(first) > len(second):
        first, second = second, first
    distances = range(len(first) + 1)
    for index2, char2 in enumerate(second):
        new_distances = [index2 + 1]
        for index1, char1 in enumerate(first):
            if char1 == char2:
                new_distances.append(distances[index1])
            else:
                new_distances.append(1 + min((distances[index1],
                                              distances[index1 + 1],
                                              new_distances[-1])))
        distances = new_distances
    return distances[-1]
```

Figure 3: Levenshtein Distance calculation

```
def build_graph(nodes):

    gr = nx.Graph() # initialize an undirected graph
    gr.add_nodes_from(nodes)
    nodePairs = list(itertools.combinations(nodes, 2))
    for pair in nodePairs:
        firstString = pair[0]
        secondString = pair[1]
        levDistance = levenshtein_distance(firstString,
secondString)
        gr.add_edge(firstString, secondString,
weight=levDistance)
return gr
```

Figure 4: Graph building using sentences

```
def extract_sentences(text, summary_length=5,
clean_sentences=False, language='english'):
    prs = Presentation()
    summ = ''
    sent_detector =
nltk.data.load('tokenizers/punkt/'+language+'.pickle'
)
    sentence_tokens =
sent_detector.tokenize(text.strip())
    graph = build_graph(sentence_tokens)
    calculated_page_rank =
nx.pagerank(graph,weight='weight')
    # most important sentences in ascending order of
importance
    sentences = sorted(calculated_page_rank,
key=calculated_page_rank.get,
                       reverse=True)
    i=0

    for sent in sentences:
        if i<summary_length:
            summ+=sent
            i = i+1
            tsl = prs.slide_layouts[1]
            slide = prs.slides.add_slide(tsl)
            subtitle = slide.placeholders[1]
            subtitle.text = sent;
        else:
            break;
    prs.save('test1216.pptx')
    return summ
```

Figure 5: Ranking and PowerPoint Generation

Fig-2 involves extraction of text from the .docx file. Fig-3 implements the method of Levenshtein distance between any two sentences. This is used to cluster sentences and help in bringing some order into the random text. Fig-4 is used to create a graph using the Levenshtein Distance as the edge weight between any two nodes. Fig-5 implements the above methods and sorts the sentences according to their ranks. After the sentences are sorted, the summary is generated and simultaneously the slides are generated, and the PowerPoint is saved, and the summary is displayed.

Fig-6 to Fig-10 show the slides which have been generated by the graph based approach.

- He is the only player to have scored one hundred international centuries, the first batsman to score a double century in a One Day International, the holder of the record for the most number of runs in both ODI and Test cricket, and the only player to complete more than 30,000 runs in international cricket.

Figure 6: Generated slide 1

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

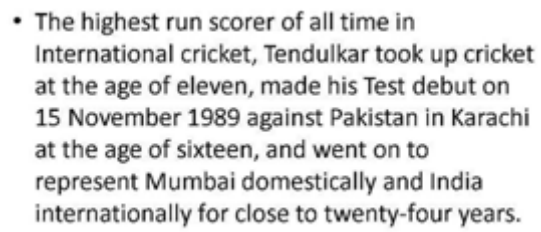

Figure 7: Generated slide 2

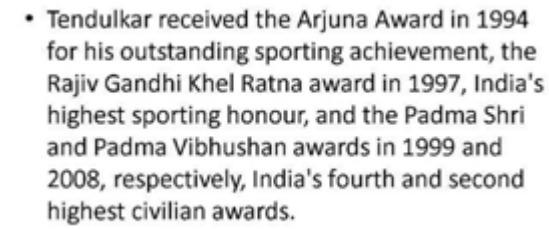

Figure 8: Generated slide 3

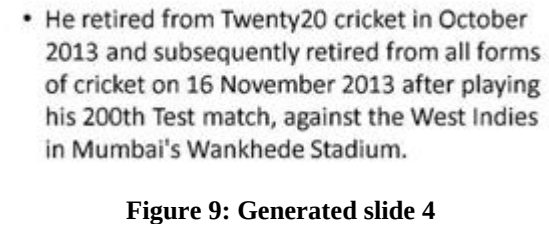

Figure 9: Generated slide 4

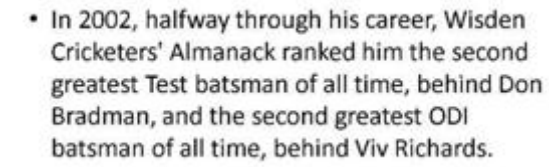

Figure 10: Generated slide 5

V. CONCLUSION

In this article, we implemented a method of generating a PowerPoint Presentation, by using a Microsoft Word File as our input. The text in the document was extracted and the sentences were summarized using the TextRank algorithm. The said summary formed the basis for the generation of the PowerPoint presentation.

REFERENCES

1. Vishal Gupta and Gurupreet Singh Lehel," A survey of text summarization extractive techniques",journal of emerging technologies in web intelligence,volume.2, no.3, August 2010.
2. Razieh Abbasi-ghalehtaki, Hassan Khotanlou and Mansour Esmaeilpour," A combinational method of fuzzy, particle swarm optimization and cellular learning automata for text summarization", journal swarm and evolutionary computation, volume 30, October 31,2016.
3. Rasim M. Alguliyev, Nijat R. Isazade,"An unsupervised approach to generating generic summaries of documents ", journal applied soft computing, volume 34 issue C, September 2015.
4. Changjian Fang,Dejun Mu,Zhenghong Deng,zhing Wu," Word-sentence co-ranking for automatic extractive text summarization "Journal Expert Systems with Applications : An international journal volume 72, issue c, April 2017.
5. Gunes Erkhan and Dragomir R. Radev,"Lexrank: graph-based lexical centralityassalienceintext summarization",journal of Artificial Intelligence research 22 (2004).
6. Hongyan Jing," Sentence reduction for automatic text summarization", proceedings of the sixth conference on applied natural language processing, association for computational linguistics Stroudsburg, 2000.
7. Lawrence H Reeve, Hyoli Han and Ari D.Brooks," The use of domain-specific concepts in biomedical text summarization", Information Processing and management, Volume 43, Issue 6, 2007.
8. Yue Hu and Xiaojun Wan," PPSGen: learning-basedpresentationslides generation for academic papers", Volume:27, issue 4, April 1,2015.
9. Ning Zhong, Yuefeng Li, and Sheng-Tang Wu," effective pattern discovery for text mining " IEEE transactions on knowledge and data engineering, volume.24, No.1, January 2012.
10. Ganesan, Kavita, ChengXiangZhai, and Jiawei Han,"Opinosis: A graphbased approach to abstractive summarization of highly redundant opinions.", Volume 2, Published 2010.
11. Massih-Reza Amini, Patrick Gallinari," Self-supervised learning for automatic text summarization by text-span extraction",23rd BCS European annual colloquium on information retrieval, 2001.
12. Priya Ganguly, Prachi M. Joshi," IPPTGen-Intelligent PPT Generator", 2016 International conference on computing, analytics and security trends (CAST) college of engineering, Pune, India. December 19-21, 2016.
13. Anusha Bagalkotkar, Ashesh Khandelwal, Shivam Pandey, Sowmya Kamath S," A novel technique for efficient text document summarization as a service", 2013 Third International Conference on Advances in Computing and Communications.
14. Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, Krys Kochut, "Text summarization techniques: A brief survey", arxiv, July 2017, USA.
15. Chaudhary M., Kumar H. (2016), "A Framework to Rank Nodes in Social Media Graph Based on Sentiment-Related Parameters", In: Satapathy S., Joshi A., Modi N., Pathak N.(eds) proceedings of international conference on ICT for sustainable development. Advances in intelligent systems and computing, Vol 409. Springer, Singapore.
16. S.A.Babara, Pallavi D.Patil, "Improving performance of text summarization", volume 46, pages 354-363,2015.
17. Yadav C.S., Sharan A., Kumar R., Biswas P. (2016) A New approach for single text document summarization. In: Satapathy S., Raju K., Mandal J., Bhateja V. (eds) Proceedings of the second international conference on computer and communication technologies. Advances in intelligent systems and computing, Vol 380. Springer, New Delhi.
18. Changjian Fang, Dejun Mu, Zhenghong Deng,Zhiang Wu" Word-sentence co-ranking for automatic extractive text summarization.", Volume 72,15 April 2017, Pages 189-195.
19. Rasmita Rautray, Rakesh Chandra Balabantaray, "Bio-inspired approaches for extractive document summarization: A comparative study", volume 3, issue 3, July 2017, Pages 119-130.
20. Farshad Kiyoumars, "Evaluation of automatic text summarizations based on human summaries", volume 192, 24 June 2015, pages 83-91.
21. Tian Wang, Yang Li, Yonghong Chen, Hui Tian, Yiqiao Cai, Weijia Jia, Baowei Wang,"Fog-based evaluation approach for trustworthy communication in Sensor-cloud system", communications letters IEEE, vol. 21, pp. 2532-2535, 2017, ISSN 1089-7798.
22. Rafael Ferreira, Luciano de Souza Cabral," Assessing sentence scoring techniques for extractive text summarization", volume 40, issue 14, 15 October 2013, pages 5755-5764.
23. Simone Teufel and Marc Moens, "Summarizing scientific articles: experiments with relevance and rhetorical status", volume 28, issue 4, December,2002 ,p.409-445.
24. [24] Ya-Han Hua, Yen-Liang Chenb, Hui-Ling Chou," Opinion mining from online hotel reviews – A text summarization approach ", Volume 53, Issue 2, March 2017, pages 436-449.
25. Francesco Ronzano, Horacio Saggion, "Knowledgeextraction and modeling from scientific publications", part of the lecture notes in computer science book series (LNCS, volume 9792).
26. M. V. K. Kiran, R. E. Vinodhini, R. Archanaa and K. Vimalkumar, "User specific product recommendation and rating system by performing sentiment analysis on product reviews," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, 2017, pp. 1-5.
27. Hans Moen, Laura-Maria Peltonen, Juho Heimonen, Antti Airola, Tapio Pahikkala, Tapio Salakoski, Sanna Salanterä, "12", Artificial Intelligence in Medicine, Volume 67, February 2016, Pages 25-37.

28. Ailin Li, Tao Jiang, Qingshuai Wang, Hongzhi Yu," The Mixture of Textrank and Lexrank Techniques of Single Document Automatic Summarization Research in Tibetan", 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 27-28 August,2016.
29. Hannah M.E., Geetha T.V., Mukherjee S. (2011) Automatic Extractive Text Summarization Based on Fuzzy Logic: A Sentence Oriented Approach. In: Panigrahi B.K., Suganthan P.N., Das S., Satapathy S.C. (eds) Swarm, Evolutionary, and Memetic Computing. SEMCCO 2011. Lecture Notes in Computer Science, vol 7076. Springer, Berlin, Heidelberg.
30. Rada Mihalcea, Paul Tarau,"TextRank: Bringing Order Into Texts", Conference on Empirical Methods, in Natural Language Processing, 2004.