Extracting Core Contents from a Research Article by Generating a Sentence Graph

Keerthi Krishnan[1], K. S. Easwarakumar[2]
[1,2]*Department of Computer Science & Engineering, Anna University, Chennai 600 025, INDIA*

Abstract **- There is a tremendous increase in the number of scientific articles available online. Hence the number of articles displayed as a result of a search query performed based on a specific research topic is also increasing. Naive researchers need to spend a lot of time reading these articles and understand the idea mentioned in each article. Also, they face difficulty in choosing appropriate articles for performing their literature search efficiently. Focusing on these concerns, an extractive summary of a research article is generated using the Core Content Extraction System proposed in this paper. Herein, the most relevant sentences from a research article are extracted. From these extracted sentences, a summary is generated which will benefit researchers by enabling them to decide whether the respective research article is useful for them or not. Experimental results show that the proposed method attain good results when compared to some existing text summarization systems.**

Index Terms **– Text Summarization; Content Extraction; Extractive Summary; Information retrieval.**

INTRODUCTION

The quantity of information available on the internet is growing exponentially. Also, the rate of doing research in various scientific domains is increasing and as a result, there is a tremendous increase in the number of research articles published every year. An interesting factor is that most of these publications are available online and are freely accessible. Researchers usually use search engines such as Google Scholar, Citeseerx, etc., or online repositories such as Digital Bibliography & Library Project (DBLP) to find research articles published on a particular research topic. They will perform the search using the required topic as their search keyword. While performing the search, they will get a list of links pointing to research articles containing the search keyword. Now the researcher needs to download each article and read its contents to decide whether the article is useful or not. A lot of time and space is been wasted in this process as it takes space to save these documents and time to read their contents. In such a scenario, there is a need for developing a system that can extract the most relevant contents from the article and generate a summary document of the same. Thus, instead of reading the entire article, researchers can read this summary and decide whether that article is useful or not. These summaries are useful for writing the "related work" section of a new article. Also, researchers can use these summaries while writing a survey paper based on a particular topic.

Text summarization is the process of automatically creating a compressed version of a given text that provides useful information for the user[1]. Mainly there are two types of text summarization: - Extractive summarization and Abstractive summarization[2]. Extractive summaries are created by extracting relevant sentences from the document to be summarized, whereas Abstractive summaries are written to convey the significant information in the input document.

Research on extractive summarization in the fifties and sixties is based on some simple features of sentences in the source document [3,4]. The Trainable Document Summarizer[5] performs sentence extraction, based on a number of weighting heuristics. Graph-based models show potential results for text summarization. The main objective of these approaches is to find the most essential sentences in the source document by constructing a graph in which nodes are sentences and edges are similar between these sentences. Such models are included in LexRank[1], and TextRank [6]. A language- and domain-independent automatic text summarization approach is proposed by Garcia – Hernandez et.al.[7] by taking out sentences using an unsupervised learning algorithm. COMPENDIUM is a text summarization system

proposed by Elena et.al.[8] for generating abstracts of biomedical papers. They present two approaches for generating summaries: $COMPENDIUM_E$ generates extractive summaries and $COMPENDIUM_{E-A}$ produces abstractive summaries respectively. The research article[9] proposes a summarization technique based on sentence clustering and shows that the summarization result depends on sentence similarity measures also. A summarization approach for scientific articles[10] is proposed which takes advantage of citation-context extracted from the reference article and the document discourse model.

A lot of authors suggest Summarization approaches based on topic modeling and Bayesian models [11, 12, 13, 14]. In these approaches, the content allocation in the final summary is projected using a graphical probabilistic model. Various approaches [15, 16] have considered summarization as an optimization task solved by linear programming and several other works have viewed the summarization problem as a supervised classification problem. Some of the supervised models utilized for summary generation are HMM [17], CRF [18], SVM [19]. Recently, extractive summarization models are developed for text in different languages [20, 21].

Another work[22] is proposed to evaluate the topic-dependent impact of scientific articles based on the modified PageRank algorithm. A content-based journal and conference recommender system is proposed [23] for computer science publications. This recommender system uses the abstract of the respective publications to suggest suitable journals and conferences. A content extraction tool, PDFdigest[24], is presented which extracts structured textual contents from scientific articles. This tool extracts textual content from articles in PDF format from both text and image-based files.

This paper proposes a Core Content Extraction System (CCES), which extracts the most relevant contents from a research article and generates an extractive summary. Researchers can use this summary to decide whether the respective article is valuable for their research. The rest of the paper is organized as follows. A detailed explanation of the proposed content extraction system is presented in section 2. The experimental framework is given in section 3 which describes the dataset, evaluation metrics and result obtained. Finally, section 4 deals with the concluding remarks and future directions.

CORE CONTENT EXTRACTION SYSTEM

The proposed system CCES extracts the most significant content from a research article and generates summarized information of the respective article. Research articles published in Computer Science (CS) field, specifically in Computational Geometry (CG) domain are considered for evaluating the system. Text portions selected from the *Abstract*, *Introduction* and *Conclusion* sections of a research article are considered for the study. This section gives a detailed description of the methodology used to identify core sentences of a research article and extract those sentences to generate a summary of the article. The various steps in CCES are shown in Figure 1 and the detailed explanation of these steps is given in the following sub-sections.

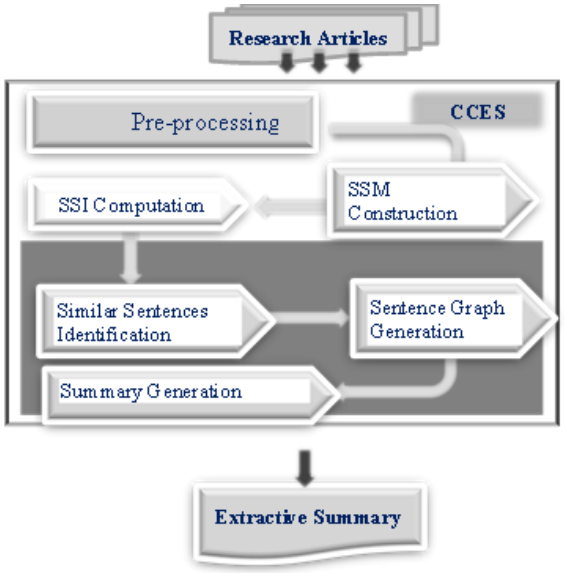

Figure 1: System Architecture of CCES

A.prereprocessing

The raw data are taken from the Abstract, Introduction, and Conclusion sections of the research article is split into sentences and are indexed according to the order in which they appear in the raw text starting from 1 till *n*, where *n* is the total number of sentences. Let S = $\{s_1, s_2, \ldots, s_n\}$ be the set of actual sentences taken from the article where s_1 is the first sentence and s_i , s_{i+1} are adjacent sentences. This set is used for generating the summary as explained in the sub-section "F" and hence numbering given to the sentences should be preserved. In this work, the punctuation symbol "." is considered as the delimiter

for identifying sentences. Various pre-processing steps such as removal of stopwords and punctuation marks are done to clean the data. As well, some irrelevant terms such as *corresponding author*, *email addresses* are also eliminated by considering them as special stopwords. Finally, a set of refined sentences $S^R = \{r_1, r_2, \ldots, r_n\}$ is created, where r_i is the refined sentence of s_i. The set S^R is used for further computation.

B. Sentence Similarity

The similarity between two sentences in a document is computed as the number of words common in those two sentences [6]. Let $W_i = \{w_1, w_2, \ldots, w_m\}$ be the set of words in a sentences s_i, where n is the total number of sentences in a document and m is the number of words in a sentences s_i. The similarity of the two sentences s_i and s_j can be computed as per equation 1.

$$similarity(s_i, s_j) = \frac{|\{w_k | w_k \in W_i \& w_k \in W_j\}|}{log(|W_i|) + log(|W_j|)} \quad (1)$$

The similarity value varies from 0 to 1. A value of 1 indicates that the two sentenceactly the same and 0 indicates that they are different. Any value in between 1 and 0 indicates the similarity between the sentences. A Sentence Similarity Matrix (*SSM*) is constructed to store the similarity values of all sentences in the document. Since there are n sentences, the size of *SSM* is *n* x *n*. *SSM*[*i*][*j*] stores the similarity value of sentences s_i and s_j is computed using equation 1.

C. Similar Sentence Index Computation

For each sentence $r_i \in S^R$, if the values stored in an i^{th} row of *SSM* from *SSM*[*i*][*1*] till *SSM*[*i*][*n*] falls above a threshold value t_s then the index of sentences similar to r_i are

$$SSI_i = \{ j \mid SSM[i][j] \geq t_s, 1 \leq j \leq n \text{ and } j \neq i \}$$

Note that there may be some SSI_i's whose cardinality may be zero. Now, SSI_i's whose cardinality falls above a threshold value are only considered for further computation.

D. Identify Similar Sentences

Let the set I_i store the final set of the index of sentences similar to $r_i \in S^R$. Initially $I_i = \{i\}$. For each $k \in SSI_i$, find the index of sentences similar to both r_i and r_k stored in SSI_{ik} .The computation of SSI_{ik} is formally written as $SSI_{ik} = SSI_i \cap SSI_k$. Next, find the index *k* such that

$$|SSI_{ik}| = L_{max} \quad (2)$$

where $L_{max} = \max \{ |SSI_{ik}| : k \in SSI_i \}$. If there is more than one *k* satisfying equation 2, then a set *K* is created such that $K = \{k: |SSI_{ik}| = L_{max}\}$ and choose the first *k* index arbitrarily. For further computation, consider those SSI_{ik} satisfying equation 2 and $k \in K$.

Now, update the set I_i by adding *k* to the set, ie, $I_i = I_i \cap k$. Repeat the above process by reassigning SSI_i to SSI_{ik} until I_i and SSI_{ik} are equal. Similarly, the index set is computed for all the sentences and the final Index set is created as $FI = \{I_1, I_2, \ldots, I_n\}$.

E. Sentence Graph Generation

A graph termed a sentence graph, is constructed from the final index set as an edge-weighted graph and is defined as follows: Let *SG (V, E, W)* denotes a Sentence Graph where *V* is the set of vertices defined as $V = FI$, *E* is the set of edges defined as $E = \{(I_i, I_j) \mid I_i \cap I_j \neq \emptyset \ \forall \ I_i, I_j \in V\}$ and *W* stores the cost of edges in *E*. The cost of edge $e_k \in E$ is defined as $c_k = |I_i \cap I_j|$ where $e_k = (I_i, I_j)$.

For each vertex $v \in V$ in *SG*, a node score, NS_v is computed as the sum of the cost of all the edges incident to that node, written formally as $NS_v = \Sigma\, c_k$. Then, choose the vertex *v* having the maximum node score as follows: $NS_v = NS_{max}$ where $NS_{max} = \max\{NS_v : v \in V\}$. If more than one vertex has the maximum node score, then choose the vertex having a minimum number of entries in the corresponding I_i and the same I_i can be considered as the index set of most similar sentences.

F. Summary Generation

Let the set *SI* contains the index of most similar sentences in the document, where $SI = I_i$. Extract the sentences whose indexes are in *SI* from the set *S* and clubbed together to generate the summary sentences *SS*.

where $SS = \{s_i \mid i \in SI \text{ and } s_i \in S\}$. Researchers can read these summary sentences and decide whether the document is useful or not.

EXPERIMENTAL FRAMEWORK

This section gives a short description of the dataset employed in the work, the evaluation metrics considered for analysis, and the results achieved.

Dataset

Research articles published in the Computational Geometry (CG) research field are considered for experimenting with the proposed system. The articles are directly collected from the homepages of various journals and conference proceedings in the CG field such as Computational Geometry - Theory and Applications (CGTA), Journal of Computational Geometry (JOCG), Discrete Computational Geometry (DCG), Canadian Conference on Computational Geometry (CCCG), Symposium on Computational Geometry (SOCG).

Initially, the collected research articles are in the Portable Document Format (PDF) and are converted to plain text for further processing. A typical research article contains the title of the paper, author(s) and their affiliations, keywords, abstract, introduction, various sections explaining the content of the article and the discussion of the results obtained, conclusion, acknowledgment (can be optional), and references. As the main idea of this paper is to extract the core contents of a research article, we feel that the summary generated by the proposed system should contain (1) the research problem focused in the article, (2) the methodology used to solve the problem and (3) the achievements obtained. Text contents from the abstract section of an article can be considered as a summary written by the authors of the article. But the fact is that all abstracts may not have the most required contents of the article, as some of them may be too precise and some others may contain unimportant contents. Processing the entire contents of the article takes too much time as it contains a detailed explanation of the methodology and discussion on results obtained. Under the assumption that in most of the articles, the *Abstract*, *Introduction*, and *Conclusion* sections cover the main idea addressed in the article, the text portions from these sections are only considered for analyzing the proposed system.

EVALUATION METRICS

The main approach for determining the quality of a summary is by content evaluation which can be done by comparing the generated summary with an ideal summary. Most of the researchers used human-generated summaries as the ideal summary due to the lack of availability of any standard summary for academic research articles. The sentences taken from the *Abstract* section of a research article can be considered as the summary written by the respective authors and hence utilize as the ideal summary in this work. In this paper, the summary generated by the proposed system is termed as *system summary* and the baseline summary used for comparison is coined as *reference summary*. Content evaluation can be done in two ways: co-selection measures and content-based measures [25].

1. co-selection measures:

The quality of extracted sentences is often measured by co-selection measures. It checks how many sentences in the reference summary are contained in the system summary. The main evaluation metrics that come under this category are precision, recall, and F-score. Precision is computed as the ratio of the number of sentences occurring in both system and reference summaries to the number of sentences in the system summary. The recall is computed as the ratio of the number of sentences occurring in both system and reference summaries to the number of sentences in the reference summary. F-score is a composite measure that combines precision and recall.

2. Content-based measures

In content-based measures, instead of comparing the entire sentences, they compare the actual words in a sentence. The main evaluation metrics under this category are ROUGE [26] scores, which calculates precision, recall, and F-measure values. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It includes measures to automatically determine the quality of a summary by comparing it to ideal summaries created by humans. The most common ROUGE metrics are: ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-SU4 where ROUGE-1, ROUGE-2 compute the number of overlapping unigrams and bigrams, respectively; ROUGE-L calculates the longest common subsequence between two summaries; and ROUGE-SU4 measures the overlap of skip-bigrams an automatic summary contains with respect to a model one, with a maximum distance of four words between them.

RESULTS AND DISCUSSION

This section presents the results obtained along with a discussion. The proposed system generates a summary for each article considered in the dataset and the quality of the generated summary is evaluated. The

results obtained for co-selection measures are tabulated in Table 1 for randomly chosen five research articles from the dataset. Table 2 gives the average values obtained for the entire dataset and its pictorial representation is given in figure 2. The performance of the proposed system is compared with an extractive summarization system COMPENDIUM$_E$ [8] and a state-of-art summarizer MS-Word Summarizer and the average values obtained for the co-selection measures are also shown in Table 2. The ROUGE scores achieved for the different summarization systems are shown in Table 3 and its pictorial representation is shown in figure 3.

Table 1: Co-selection Results

Research Article No.	Precision	Recall	F measure
1	0.7	0.77	0.733
2	0.83	0.45	0.58
3	0.63	0.56	0.59
4	0.38	0.6	0.47
5	0.60	0.43	0.50

Table 2: Comparison with other summarization systems

System	Average Precision (%)	Average Recall (%)	Average F measure (%)
Proposed	63.67	60.67	62.28
COMPENDIUM$_E$	51.23	47.68	46.67
MS-Word Summarizer	50.17	45.23	47.73

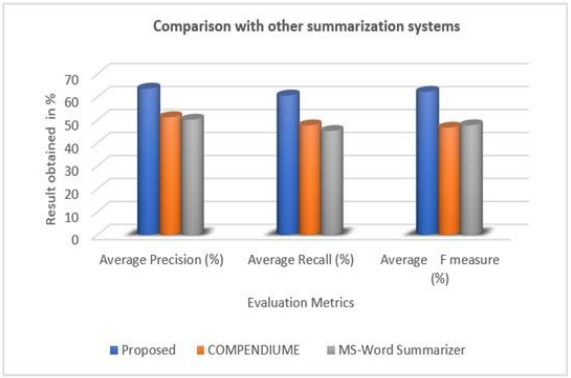

Fig. 2 Comparison with other summarization systems

These results show that the proposed system performs well when compared to the other summarization systems. The summary generated by this system is useful for researchers as they get an idea about the main contents discussed in the respective research article without reading the entire contents and can save a lot of time.

Table 3: ROUGE results

ROUGE metrics	System	Average Recall (%)	Average Precision (%)	Average F-measure (%)
ROUGE-1	Proposed	61.23	65.17	63.14
	COMPENDIUM$_E$	48.34	45.12	46.67
	MS-Word Summarizer	49.27	43.11	45.98
ROUGE-2	Proposed	42.65	44.38	43.50
	COMPENDIUM$_E$	17.56	16.41	16.97
	MS-Word Summarizer	16.22	15.28	15.74
ROUGE-L	Proposed	45.31	42.89	44.07
	COMPENDIUM$_E$	31.11	27.45	29.17
	MS-Word Summarizer	30.21	28.12	29.13
ROUGE-SU4	Proposed	32.34	33.78	33.04
	COMPENDIUM$_E$	19.12	17.40	18.22
	MS-Word Summarizer	17.24	16.54	16.78

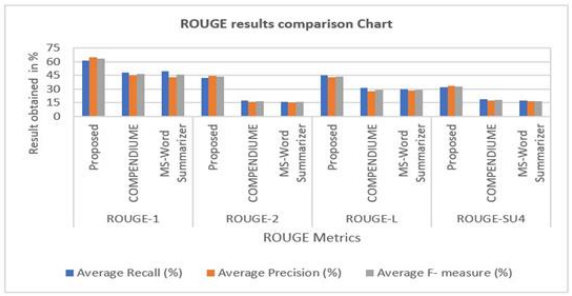

Fig. 3 Comparison with ROUGE results

The outcome of the proposed system is also analyzed based on the size of the input data along with the generated summary. Here, the size of the data is measured in terms of the number of words contained in the *Abstract*, *Introduction*, and *Conclusion* sections of the research article. Also, the number of words in the system summary and reference summary is taken into consideration for the analysis. Table 4 shows the size of the research articles listed in Table 1 along with the size of generated summary and reference summary.

Table 4: Input and output data size

Article-No.	Size of input data	Size of Reference summary	Size of System summary
1	2756	257	296
2	1466	296	150

3	2316	127	124
4	641	76	164
5	1961	143	114

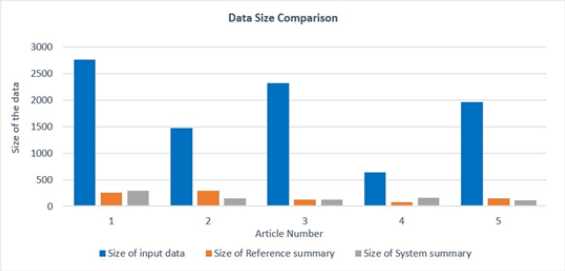

Fig. 4 Comparison with the size of input data and output summary

Here, the size of the summary generated by the proposed system is varying when compared to the reference summary. The graphical representation of this comparison along with the size of the input data is shown in figure 4. This variation shows that the output of the proposed system extracts sentences not exactly from the *Abstract* section but also the other sections considered in the input. This leads to the fact that the generated summary will be containing better facts than that of the paper abstract.

CONCLUSION

A framework for extracting the core contents from a research article is proposed in this paper and an extractive summary is been generated from the extracted contents. These summaries are useful for researchers as they get an overview of the idea discussed in the research article without reading the entire contents. These summaries are also useful for writing the *related work* section of a new research paper. The quantitative results show that the proposed core content extraction system was able to generate good summaries. As future work, the extracted sentences can be used for generating a survey paper for a particular research topic.

REFERENCES

[1] Erkan, G.; Radev, D.R. "Lexrank: Graph-based lexical centrality as salience in text summarization", Journal of artificial intelligence research, Vol *22*, pp 457-479, 2004.

[2] Nenkova, A.; McKeown, K. "Automatic summarization", Foundations and Trends® in Information Retrieval, Vol 5, No.2–3, pp 103-233, 2011.

[3] Luhn, H. P. "The automatic creation of literature abstracts", IBM Journal of research and development, Vol 2, No.2, pp 159-165, 1958.

[4] Edmundson, H. P. "New methods in automatic extracting*"*, Journal of the ACM (JACM*)*, Vol 16, No.2, pp 264-285, 1969.

[5] Kupiec, J.; Pedersen, J.; Chen, F." A trainable document summarizer", In Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 68-73, July 1995.

[6] Mihalcea, R.; Tarau, P. "Textrank: Bringing order into text", In Proceedings of the 2004 conference on empirical methods in natural language processing, 2004.

[7] García-Hernández, R. A.; Montiel, R.; Ledeneva, Y.; Rendón, E.; Gelbukh, A.; Cruz, R. "Text summarization by sentence extraction using unsupervised learning",In Mexican International Conference on Artificial Intelligence Springer, Berlin, Heidelberg, pp. 133-143, October 2008.

[8] Lloret, E.; Romá-Ferri, M. T.; Palomar, M. "COMPENDIUM: A text summarization system for generating abstracts of research papers", Data & Knowledge Engineering, Vol 88, pp 164-175, 2013.

[9] Zhang, P. Y.; Li, C. H. "Automatic text summarization based on sentences clustering and extraction", In Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology, pp. 167-170, August 2009.

[10] Cohan, A.; Goharian, N. "Scientific article summarization using citation-context and article's discourse structure", arXiv preprint arXiv:1704.06619, 2017.

[11] Celikyilmaz, A.; Hakkani-Tur, D. "A hybrid hierarchical model for multi-document summarization", In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 815-824, July 2010.

[12] Vanderwende, L.; Suzuki, H.; Brockett, C.; Nenkova, A. "Beyond SumBasic: Task-focused summarization with sentence simplification and

lexical expansion", Information Processing & Management, Vol 43, No. 6, pp 1606-1618, 2007.

[13] Ritter, A.; Cherry, C.; Dolan, B. "Unsupervised modeling of twitter conversations In Human Language Technologies", The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 172-180, June 2010.

[14] Li, J.; Li, S. "A novel feature-based bayesian model for query focused multi-document summarization", Transactions of the Association for Computational Linguistics, Vol 1, pp 89-98, 2014.

[15] Clarke, J.; Lapata, M. "Global inference for sentence compression: An integer linear programming approach", Journal of Artificial Intelligence Research, Vol 31, pp 399-429, 2008.

[16] Woodsend, K.; Lapata, M. "Multiple aspect summarization using integer linear programming", In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 233-243, July 2012.

[17] Conroy, J. M.; Schlesinger, J. D.; Kubina, J.; Rankel, P. A.; O'Leary, D. P. "CLASSY 2011 at TAC: Guided and Multi-lingual Summaries and Evaluation Metrics', pp 1-8, 2011.

[18] Chali, Y.; & Hasan, S. A. "Query-focused multi-document summarization: Automatic data annotations and supervised learning approaches", Natural Language Engineering, vol 18, no 1, pp - 109-145, 2012.

[19] Xie, S.; Liu, Y. "Improving supervised learning for meeting summarization using sampling and regression.", Computer Speech & Language, Vol, 24, No 3, pp 495-514, 2010.

[20] Nawaz, A.; Bakhtyar, M.; Baber, J.; Ullah, I.; Noor, W.; & Basit, A. "Extractive Text Summarization Models for Urdu Language", *Information Processing & Management*, Vol *57* (6), 102383, 2020.

[21] Abu Nada, A. M.; Alajrami, E.; Al-Saqqa, A. A.; & Abu-Naser, S. S. "Arabic Text Summarization Using AraBERT Model Using Extractive Text Summarization Approach"., 2020

[22] Zhang, Y.; Ma, J.;Wang, Z.; Chen, B.; Yu, Y. "Collective topical PageRank: a model to evaluate the topic-dependent academic impact of scientific papers", Scientometrics, Vol 114, No 3, pp 1345-1372, 2018.

[23] Wang, D.; Liang, Y.; Xu, D.; Feng, X.; Guan, R." A content-based recommender system for computer science publications", Knowledge-Based Systems, Vol *157*,pp 1-9, 2018.

[24] Ferrés, D.; Saggion, H.; Ronzano, F.; Bravo Serrano, À. (2018)." PDFdigest: an adaptable layout-aware PDF-to-XML textual content extractor for scientific articles", In Language Resources and Evaluation Conference (LREC), pp 7 – 12, May 2018.

[25] Steinberger, J.; Ježek, K. "Evaluation measures for text summarization", Computing and Informatics, Vol 28, No 2, pp 251-275, 2012.

[26] Lin, C. Y. "Rouge: A package for automatic evaluation of summaries", Proceedings of the Association for Computational Linguistics Text Summarization Workshop, pp 74–81, 2004