Automatic Text Summarization using ChainRank: A Hybrid Technique

Michael M. Angeles, Nathaniel S. Laxina,
Maria Art Antonette D. Clariño and Rizza DC. Mercado
Institute of Computer Science
University of the Philippines Los Baños
College, Laguna, Philippines 4031
{angelesmichael21, nathaniel.laxina, maanclarino, rdcmercado}@gmail.com

ABSTRACT

This paper presents an automatic text summarizer for English articles that uses ChainRank - a hybrid approach to text summarization based on combining Lexical Chains and PageRank. Given a text document, the system extracts the essential sentences that capture the whole idea of the document according to an evaluation process where sentences are scored. Chains are used to filter sentences that will be used by the PageRank method to comprise the summary. The system also implemented abstraction by sentence reduction, generalization and condensation to yield more compressed summaries. A data set of 567 English electronic articles was requested from the Document Understanding Conference (DUC). On this data set, the performance of the system was evaluated using ROUGE-1 (n-gram(1,1)) F-measure and compared to the Top 5 of the 15 systems that participated in DUC 2002.

Categories and Subject Descriptors

I.3.1 [**Artificial Intelligence**]: Natural Language Processing – *natural language generation, lexical semantics, language resources.*

General Terms

Algorithms, Performance, Languages

Keywords

Natural Language Processing, Automatic Summarization, PageRank, Lexical Chains

1. INTRODUCTION

Summarization, in its basic sense, is acquiring important information from a lengthier source. It has become an essential part of everyday life [17]. People are updated on world events by reading news articles, make investment decisions based on stock market updates, and even choose a movie based on reviews they have read. With accessible and useful summaries, effective decisions are made in less time.

As the amount of information continues to grow, it becomes an overwhelming task to filter them all. With fast-paced lifestyle, comprehending lengthy sources has become impractical. Although there are existing tools for summarization, generating effective summaries is still a problem. They do not capture the real thought of the source document. This rapid growth of information can also be said in the field of research as seen by the online accessibility of scientific literatures [19]. Apart from scholarly work, business related transactions in the form of e-commerce and posting of product reviews are performed over the Web making automatic summarization a complementary tool. Gathering of product reviews will provide a good feedback mechanism in business as demonstrated by a web-based review summarization system [32].

Automatic Text Summarization is a branch of Natural Language Processing (NLP) that deals with the automated generation of a shortened version of a text. NLP is a branch of computer science and linguistics that uses computational methods to investigate, and to model mechanisms for the understanding and production of written human language by a computer [26].

Methods to text summarization can be classified as either extractive or abstractive. Summarization by extraction performs by selecting important sentences from the original document and presenting them together as a summary [13]. On the other hand, the process of abstraction works as how humans summarize by analyzing the content of the source document and breaks it down to separate parts for further processing. Although the process of abstraction tries to present summaries more readable, it is not a well-developed field yet [9]. It is still considered a challenge and requires further exploration [20]. Users prefer extractive summaries because they present information as-is by the author. Moreover, [13] mentioned that one of the problems with abstractive methods is sentence synthesis. Abstractive summaries that are generated often result in incoherence even in sentence level. Other problems include semantic analysis and natural language generation [10].

This paper presents ChainRank, a new extractive approach to text summarization based on combining Lexical Chains and PageRank, for an automatic text summarization of electronic English documents. Specifically, it must be able to extract sentences from a given document that capture its main idea, apply lexical chains and PageRank, further compress the extracted sentences by performing abstractive methods, and compare the performance (in terms of F-measure) of the developed summarizer against other systems from DUC.

2. REVIEW OF LITERATURE

As early as 1950s, several studies have been conducted concerning the development of automatic summarization systems [19]. Researchers have recognized the need for these systems for efficient information gathering. Data mining related studies have expressed importance of a document abstract or summary. Information retrieval is dependent on the presence of document summary or abstract. Its absence makes the ranking of documents in the proposed system in [16] not possible.

An early approach to extractive summarization can be seen in [8], where sentence selection was based on its location on the document and the number of cue words, key words, and title words it contains. Scores are given to sentences based on these features and the top-scoring sentences are included in the summary. This approach is considered classical in [13] because it forms the foundation of extractive methods today as seen in [11]. A trainable summarizer was presented taking in several features such as sentence position, positive keyword, negative keyword, sentence centrality, sentence resemblance to the title, sentence inclusion of numerical data, sentence relative length, bushy path of the sentence, and aggregated similarity for each sentence to generate summaries.

There are limitations in using sentence location and cue words. The use of those features is dependent on the type of document. Because of this, [3] introduced an approach that uses lexical chains. Chains are scored based on length and relationships of their members. Strong chains or those with scores greater than a given threshold will comprise the summary. This approach relies on the content and not on the structure, thus independent of the type of document.

Another study described an approach for identifying the most important parts of the text which are topically most salient [5]. The approach takes into consideration the connectedness of sentences. It efficiently uses lexical chains and employs the use of a text segmenter to get segments of the text that address the same topic. This particular study suggested the use of compression techniques to increase the condensation of the summary. Sentences are ranked based on rules and the highest scored are considered to be representing the "true sense" of the text. Thus, lexical chains can identify the main theme of texts [33]. This was utilized in a semantic approach to text clustering stating the need for an efficient text clustering in terms of performance and data size and more importantly successfully represent the text topic [33].

In 2004, [25] introduced TextRank, a graph-based ranking model for extractive summarization. In this approach, each sentence from a given document is treated as a vertex on a graph. A link between any two sentences will be established if they share common words. Sentences are scored depending on the number of links they have and the score of the sentences they are linked to. The scores will determine its inclusion in the summary. In the same year, [10] introduced LexRank, also a graph-based approach. LexRank is based on the concept of eigenvector centrality in the graph representation of sentences for computing each sentence's importance. TextRank and LexRank were based on PageRank, a link-analysis algorithm, which is used by the Google search engine in filtering web pages that are relevant to the query of a user. The two approaches differ mainly on their computation of sentence similarity.

Recent paper showed the need for higher-order lexical models and the weakness of using PageRank. According to [12], the process of making associations by random walk to convergence in PageRank may lead to semantic drift. This means small numbered direct associations may lead to indirect associations that are not applicable. An example presented in [12] is the possible association of *breakfast venues* to *soccer fields* if associations such as *breakfast – pancakes*, *pancakes – hashbrowns*, *hashbrowns – potato*, and *potato - field* are made. These indirect associations are crucial in a question-answering study such as in [12]. This perceived weakness of PageRank was also observed in [6]. The direct and underlying links in multi-threaded chat conversations were ranked using PageRank. It was less powerful in terms of accuracy as compared to other systems presented in [6]. However, one of the advantages in using PageRank is its speed and can be applied to real-time processing.

The methods mentioned are some of the basic approaches, together with their dependencies and weaknesses, developed in the field of extractive summarization. To achieve greater results, basic techniques are often combined to create hybrid approaches which form the foundation of the most advanced systems today.

3. METHODOLOGY

3.1 Materials and Tools Used

Data Set - requested from the Document Understanding Conference (DUC) 2002 [7]. It is a large collection of documents consisting of 567 English articles - journals, business, finance, economic and political news articles. The same data set was utilized in evaluating graph-based ranking approaches such as TextRank [25] and in [31], which assumed documents within document set are related to one another in order to consider cross-document relationships. More recent studies in 2015 such as [30] utilized DUC 2002, together with DUC 2001 and 2004, for training and testing as well as in [20] that considered DUC 2002 as standard corpus in the field of summarization.

Stanford Parser - an open-source parser capable of parsing and analyzing the grammatical structure of sentences [21]. A list of stop words, obtained from ROUGE 1.5.5, was used for the removal of stop words. For the stemming of words, the Porter Stemming Algorithm was used [27].

ROUGE 1.5.5 - a toolkit implemented in PERL programming language for evaluating automatically generated summaries [22]. This tool is greatly associated with DUC performance evaluation [32].

A comparative study on commercial tools for text summarization utilized ROUGE tool for comparing the results using n-gram (n=1) configuration applied on DUC 2002 data set [14]. In a 2016 paper, the performance of [1] on the same data set using ROUGE-1 and ROUGE-2.

WordNet 3.0 - a large database of nouns, adjectives, verbs, and adverbs, was used for the retrieval of the meanings and relationships between words [34].

Java API for Wordnet Searching (JAWS) - a Java API for retrieving data in WordNet [29].

Java Orthography (JOrtho) - an open-source spell-checker implemented in Java. This tool provides the user suggestions for misspelled words in real-time [18].

3.2 Process Flow

ChainRank is an extractive approach to text summarization which is based on lexical chains and the PageRank algorithm. The system works by creating chains of nouns which are related with each other and using them as an essential factor on the scores of sentences computed based on the PageRank algorithm. Extraction of the highest scoring sentences is then performed. The system applies abstraction by reducing, generalizing, and condensing the sentences produced by extraction. Results generated by the system were compared to summaries provided in the Document Understanding Conference (DUC) using the ROUGE evaluation toolkit. The major processes in this hybrid approach are shown in Figure 1.

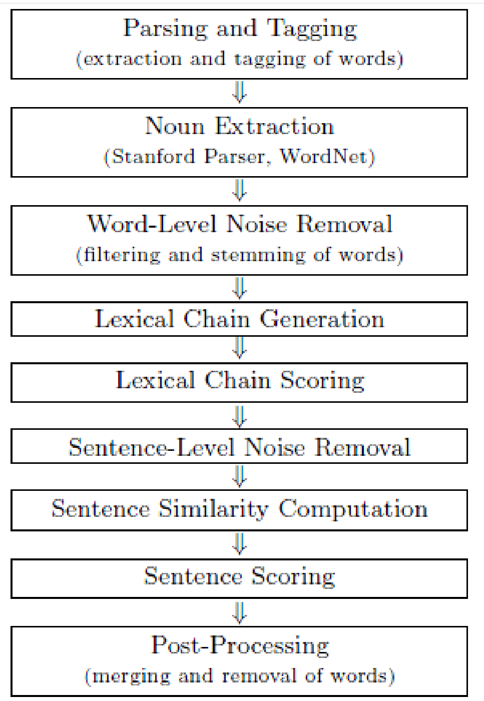

Parsing and Tagging
(extraction and tagging of words)
Noun Extraction
(Stanford Parser, WordNet)
Word-Level Noise Removal
(filtering and stemming of words)
Lexical Chain Generation
Lexical Chain Scoring
Sentence-Level Noise Removal
Sentence Similarity Computation
Sentence Scoring
Post-Processing
(merging and removal of words)

Figure 1.Overview of the processes in ChainRank

3.2.1 Parsing

Parsing is the first step of the summarization. The given document is parsed into paragraphs, paragraphs into sentences, and sentences into words. The tagging of each word in each sentence with its corresponding part of speech is included in this step.

3.2.2 Noun Extraction

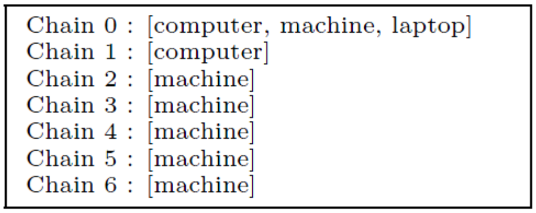
Words classified as nouns by the previous step are collected and stored in an array. Noun compounds are identified by checking if they have entries in the WordNet database.

3.2.3 Word-level Noise Removal

Noise are removed by filtering words through removal of stop words such as articles ('the', 'a', 'an'), prepositions, among others. This step is important to eliminate words that contribute little/none to the meaning of the resulting summary. The remaining words are then stemmed to obtain the root of each word.

3.2.4 Lexical Chain Generation

The extracted nouns are used for the generation of lexical chains. Each noun is put into its senses based from the database of WordNet. Nouns that are related to each other are grouped into chains sharing the same underlying concept. In this way, chains capture the cohesion present within the text [16]. Two nouns are related if they have identical, synonym, hypernym, hyponym, or sibling relation with each other. In addition, nouns that have no entry in WordNet were considered and identified by identical relation. Noun compounds were also determined by searching for patterns on the tags produced by the parser. Below are examples of lexical chains formed from the nouns computer, machine and laptop:



Chain 0 : [computer, machine, laptop]
Chain 1 : [computer]
Chain 2 : [machine]
Chain 3 : [machine]
Chain 4 : [machine]
Chain 5 : [machine]
Chain 6 : [machine]

Figure 2.Generated Lexical Chains

As shown in Figure 2, seven (7) chains were formed: two for "computer" and five for "machine". Each chain is a different interpretation despite looking as duplicates. First chain resulted from hypernym- and hyponym- based relationship across the 3 words. Membership of each sense/word depends on its relationship with the member/s in the chain.

3.2.5 Chain Scoring

Chains that were generated are scored based on the relations of its members. Table 1 shows the scoring system used for the relations. These values were acquired using empirical testing in [28] and were subjected to the algorithm in the same study resulting to good results.

Table 1.Scores of Relations [28]

	One-Sentence	Three-Sentences	Same Paragraph	Default
Identical	1.0	1.0	1.0	1.0
Synonym	1.0	1.0	1.0	1.0
Hypernym/ Hyponym	1.0	0.5	0.5	0.5
Sibling	1.0	0.3	0.2	0.0

Hypernym and hyponym relations were limited up to 3 levels on the WordNet graph since two words are not considered relevant enough if the distance between their relations in the graph is far. Words are disambiguated by determining the chain to which it contributes most thus leaving each word to belong to exactly one chain. Scored chains that are greater than the mean of the overall score of chains are retained and used for filtering the sentences that will be included in the summary.

3.2.6 Sentence-level Noise Removal

Sentences are selected depending on their overlap with the generated lexical chains. With this measure, sentences that discuss the prevailing topic in the document are selected. Those that are not selected are considered as noise or not contributing mainly to the central idea of the document.

3.2.7 Sentence Similarity Computation and Sentence Scoring

Sentence similarity computation is done after the filtering of sentences. Each sentence is compared to every other sentence. For every comparison, the similarity is computed using the formula [24]:

$$similarity(S_i,S_j)=\frac{1}{2}(A(S_i,S_j)+B(S_i,S_j))$$

$$A(S_i,S_j)=\frac{\sum_{w\in S_i} isContained(w,S_j)*idf(w)}{\sum_{w\in S_i} idf(w)}$$

$$B(S_i,S_j)=\frac{\sum_{w\in S_j} isContained(w,S_i)*idf(w)}{\sum_{w\in S_j} idf(w)}$$

where *isContained(w, Sn)* returns 1 if the word *w* occurs in sentence *n*, 0 otherwise. The inverse document frequency is defined by:

$$idf_i=\log(\frac{N}{n_i})$$

where *N* is the total number of sentences in the document and n_i is the number of sentences where word *i* appears. The obtained sentence similarity scores are used as weights in the implementation of PageRank algorithm. The PageRank Formula is applied to obtain the score of each sentence. The calculation was performed for (30) iterations as performed in [2], [15] and [23]. The PageRank Formula is given by [4]:

$$PR^W(V_i)=(1-d)+d*\sum_{V_j\in \mathrm{In}(V_i)}\left(w_{ji}\frac{PR^W(V_j)}{\sum_{V_k\in \mathrm{Out}(V_j)} w_{kj}}\right)$$

where *d* is the damping factor set to 0.85 as implemented in [15] and [2]. In [23], damping value of 0.95 was used. It is usually set within the range of 0.85-0.95 [2] [23]. *In*(V_n) is the set of vertices that points to V_n, and *Out*(V_n) is the set of vertices that V_n points to. However, the graph used was an undirected weighted graph, which means *In*(V_n) and *Out*(V_n) have the same value. Final scores of sentences were computed based on their scores from the formula and overlap with the chains generated.

3.2.8 Post-Processing

There is a need to perform post-processing on summaries generated by extractive methods to resolve possible incoherence by fusing information, making revisions and ordering sentences [19]. The method of extraction selects a subset of sentences from a given document and the selected sentences are usually still not condensed. They can be further compressed by performing abstraction on the extracted sentences. In this study, top scoring sentences identified by the algorithm are selected. Further compression is performed by merging and removing words and sentences of the top scoring sentences. Sentence combination is done by checking if sentences refer to the same subject. However, only simple structures of sentences were considered. For the post-processing, the system supports the following:

Removal of Sentence Fragments - removes incomplete sentences or those that do not convey a complete thought.
Input: Nathaniel is.
Output:

Removal of Appositives - removes phrases or clauses which gives more information about other nouns.
Input: Nathaniel, that boy over there, is eating an apple.
Output: Nathaniel is eating an apple.

Removal of Determiners - removes noun modifiers such as articles, demonstratives, etc.
Input: Nathaniel is eating the apple.
Output: Nathaniel is eating apple.

Removal of Adjectives - removes words that describe other nouns or pronouns.
Input: Nathaniel is eating the delicious apple.
Output: Nathaniel is eating the apple.

Merging of Subjects - merges subjects that are siblings or holonyms in the WordNet database.
Input: Apple and banana are Nathaniel's breakfast.
Output: Edible fruits are Nathaniel's breakfast.

Merging of Direct Objects - merges direct objects that are siblings or holonyms in the WordNet database.
Input: Nathaniel is eating apple and banana.
Output: Nathaniel is eating edible fruits.

Merging of Sentences - merges sentences that have the same subjects.
Input: Nathaniel is eating apple. Nathaniel is drinking water.
Output: Nathaniel is eating apple and drinking water.

4. RESULTS AND DISCUSSION

The performance of the summarizer was evaluated using the default setting (95% confidence interval) of ROUGE 1.5.5 evaluation toolkit. Only the first 100 words of the generated summaries were considered. The F-measure from the ROUGE-1 (n-gram(1,1)) scores was used as the metric of evaluation. F-measure is the average of recall and precision. Recall is the ratio of system-human overlap sentences over the sentences chosen by the system while precision is with respect to the sentences chosen by a human.

Table 2. Results of Improvements for ChainRank

Improvement	ROUGE-1 F-Measure Results
Default	0.43161
Stemmed	0.43415
Stopped	0.44107
Stemmed & Stopped	0.44620
With Sentence Filter	0.44753
With Chain Scores	0.44755
Combined	**0.45139**

For this study, the performance of ChainRank is compared to the improvements done to achieve better results and not directly to Lexical Chains and PageRank. Table 2 shows the improvements applied to ChainRank and their respective performance measure. The default ChainRank, which is already based on combining lexical chains and PageRank, only resulted to 0.43161. In order to rank well against the top systems in DUC 2002, different improvements had to be explored and implemented finally arriving with the combination of the earlier improvements garnering the highest F-Measure. Word-level noise removal, especially by removing the stop words, significantly improved the performance of the system. Though slight improvements were achieved by separately using sentence filtering and chain score addition, using them together further improved the performance of the system.

An example of a good summary generated in this study is shown below with numbered sentences and extracted nouns:

(1) *Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas.* (2) *The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.* (3) *The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a ``broad area of cloudiness and heavy weather'' rotating around the center of the storm.* (4) *Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet feet to Puerto Rico's south coast.*

For this resulting summary, it demonstrated how lexical chains worked to find sentences having the context of hurricane, winds, rains, and storm (which is the focus of the news article). Furthermore, lexical chains and sentence similarity (nouns) scores that were fed to the PageRank algorithm were effective in determining which sentences are the most important.

On the contrary, below is an example of a "bad" summary based on ChainRank output: D062.P.100.J.1.AP891018-0301.html:

(1) *Most San Francisco-area homeowners may have to pay for damage from Tuesday's earthquake out of their own pockets, while insurance companies may reap long-term benefits from higher rates, industry spokesmen and analysts said Wednesday.* (2) *Only 15 percent to 20 percent of California homeowners have earthquake insurance, which typically requires a 10 percent deductible and costs between $200 to $400 a year for a $100,000 home, according to industry spokesmen.* **(3)** *Industry analysts predicted insurers would be able to reverse three years of declining rates and win rate hikes from state regulators due to the quake damages and the estimated $4 billion in damages from Hurricane Hugo, which hammered South Carolina and other parts of the southeastern United States earlier this month.*

Long sentences have high possibility of being included in the final summary since these sentences will be scored high during lexical chains generation and PageRank algorithm. In the summary generated by ChainRank above, sentence **(3)** (contains 51 words, sentence with highest number of words in the article) was included in the final summary; average number of words per sentence is 20 in the article. In this example, long sentences that do not describe the essential/most relevant information of the article/document have high possibility of being included in the final summary.

The word database greatly affected the performance of the developed algorithm as it is the basis of the noun compound identification. Another factor having an effect on the result is the grammatical correctness of the input for the tagging of words. Misspelling of words can also cause problem. To lessen its occurrence, the system is incorporated with a spell checker, JOrtho that provides the user suggestions in real-time for a misspelled word.

To provide an objective evaluation in this study, the summarized data set obtained from DUC was used as benchmark just as other systems did. This makes it an ideal arena for comparison. At this point, the only interest is how the developed summarizer will rank against other systems. Therefore, only data set for evaluation was requested from DUC and no information about other participating systems was obtained. Table 3 shows the performance of ChainRank as compared to the Top 5 of the 15 participating systems in DUC 2002 as how other summarizing techniques in [25] and [31] were also evaluated.

Table 3.Scores of ChainRank and Top 5 systems that participated in DUC 2002

System	ROUGE-1 F-Measure Results
S28	0.46499
S19	0.45914
S21	0.45801
ChainRank	**0.45139**
S29	0.45057
S27	0.44770

Due to the scope of the data, the algorithm was modified to make it more suited with news articles. The system showed an improvement from 0.45139 to 0.45382 (still in the same position as compared to Table 3 results).

Because of not having any details on the approaches of the participating systems, the discrepancies on the ROUGE-1 F-Measure results found in Table 3 cannot be further discussed. However, the result satisfies the objective of evaluating the hybrid technique against these systems.

A user interface (Figure 3) was also created to execute ChainRank on smaller set of documents and automatically generate their summaries.

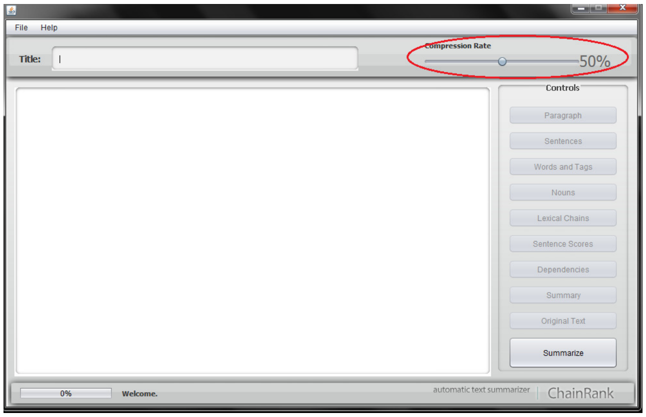
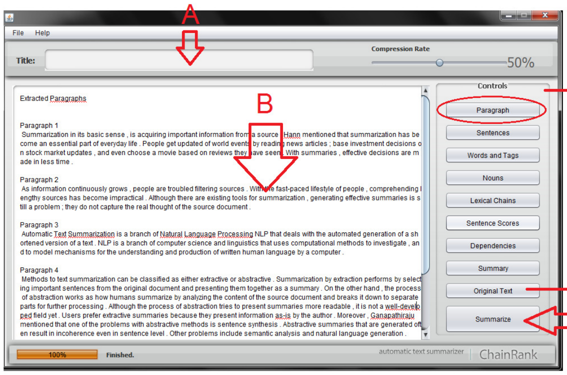


Figure 3.User interface with compression slider for specifying compression rate of summary



Extracted Paragraphs

Paragraph 1
Summarization in its basic sense , is acquiring important information from a source . Hann mentioned that summarization has become an essential part of everyday life . People get updated of world events by reading news articles ; base investment decisions on stock market updates , and even choose a movie based on reviews they have seen . With summaries , effective decisions are made in less time .

Paragraph 2
As information continuously grows , people are troubled filtering sources . With the fast-paced lifestyle of people , comprehending lengthy sources has become impractical . Although there are existing tools for summarization , generating effective summaries is still a problem ; they do not capture the real thought of the source document .

Paragraph 3
Automatic Text Summarization is a branch of Natural Language Processing NLP that deals with the automated generation of a shortened version of a text . NLP is a branch of computer science and linguistics that uses computational methods to investigate , and to model mechanisms for the understanding and production of written human language by a computer .

Paragraph 4
Methods to text summarization can be classified as either extractive or abstractive . Summarization by extraction performs by selecting important sentences from the original document and presenting them together as a summary . On the other hand , the process of abstraction works as how humans summarize by analyzing the content of the source document and breaks it down to separate parts for further processing . Although the process of abstraction tries to present summaries more readable , it is not a well-developed field yet . Users prefer extractive summaries because they present information as-is by the author . Moreover , Ganapathiraju mentioned that one of the problems with abstractive methods is sentence synthesis . Abstractive summaries that are generated often result in incoherence even in sentence level . Other problems include semantic analysis and natural language generation .

Figure 4.Screenshot showing parts of the interface for input and output purposes

Figure 4 shows the different interactions the user has with the developed interface. The user can specify the title in the text field (A), input the text to summarize in the text area (B), generate the summary by clicking button (C) and display extracted paragraphs, sentences, words and tags, chains, scores, dependencies and the actual summary by clicking from (D). The buttons in (D) are only enabled once a document has been submitted and processed after clicking (C). After clicking any button in (D), the output will be viewed in the text area (B) overwriting its previous content.

5. CONCLUSION AND FUTURE WORK

An automatic text summarization system using ChainRank has been implemented. The system's performance was evaluated using the data set provided by DUC and showed that it performed well as its ROUGE-1 F-Measure score was compared against the Top 5, ranking 4th, of the 15 systems that participated in DUC 2002.

Upon evaluation of the system, a possible extension to the study is to use a wider data set to further evaluate the performance of the system. The performance of ChainRank may also be compared to systems that participated in DUC 2003-2007 and possibly other more recent extractive summarizers. For an objective comparison, the data sets used by these newer extractive summarizers must also be used. Aside from using lexical chain overlap for the sentence filtering stage, other measures may also be considered to improve its performance. The post-processing stage may be further improved to support complex approaches, especially on merging of sentences.

6. ACKNOWLEDGEMENT

The authors would like to express their deepest gratitude to the Document Understanding Conference (DUC) for providing data for summarization and comparison of results.

7. REFERENCES

[1] Abbasi-ghalehtaki, R., Khotanlou, H., and Esmaeilpour, M. 2016. Fuzzy evolutionary cellular learning automata model for text summarization. *Swarm and Evolutionary Computation*.

[2] Agirre, E. and Soroa, A. 2009. Personalizing pagerank for word sense disambiguation. *In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 33-41. Association for Computational Linguistics.

[3] Barzilay R. and Elhadad M. 1997. Using lexical chains for text summarization. *ACL/EACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

[4] Brin, S. and Page, L. 2012. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks, 56(18)*, pp.3825-3833.

[5] Brunn, M., Chali, Y., and Pinchak, C.J. 2001. Text summarization using lexical chains. *In Proceedings of Document Understanding Conference*. Citeseer.

[6] Chiru, C.G., Rebedea, T., and Erbaru, A. 2014. Using PageRank for Detecting the Attraction between Participants and Topics in a Conversation. In *WEBIST (1)*, pp. 294-301.

[7] DUC. 2002. Document Understanding Conference 2002. National Institute of Standards and Technology, U.S. Department of Commerce. http://duc.nist.gov.

[8] Edmundson, H.P. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)*,*16*(2), pp.264-285.

[9] Endres-Niggemeyer, B., Stadtweg, R., and Endres, B. 2000. Human-style WWW summarization.*Report. Hannover: University of Applied Sciences and Arts*.

[10] Erkan, G. and Radev, D.R. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, pp.457-479.

[11] Fattah, M.A. and Ren, F. 2008. Automatic text summarization. *World Academy of Science, Engineering and Technology*, *37*, p.2008.

[12] Fried, D., Jansen, P., Hahn-Powell, G., Surdeanu, M., and Clark, P. 2015. Higher-order lexical semantic models for non-factoid answer reranking. *Transactions of the Association for Computational Linguistics, 3*, pp.197-210.

[13] Ganapathiraju, K., Carbonell, J., and Yang, Y. 2002. Relevance of Cluster size in MMR based Summarizer: A Report 11-742: Self-paced lab in Information Retrieval.

[14] García-Hernández, R.A., Ledeneva, Y., Mendoza, G.M., Dominguez, Á.H., Chavez, J., Gelbukh, A., and Fabela, J.L.T. 2009. Comparing commercial tools and state-of-the-art methods for generating text summaries. *In Artificial Intelligence, 2009. MICAI 2009. Eighth Mexican International Conference on*, pp. 92-96. IEEE.

[15] Goikoetxea, J., Agirre, E., and Soroa, A. 2014. Exploring the Use of Word Embeddings and Random Walks on Wikipedia for the CogAlex Shared Task. *COLING 2014*, p.31.

[16] Gonnade, P., Bongade, S., and T. Mendhe. 2014. Information Extraction Using Text Mining by Keyword Ranking and Scoring. *International Journal of Computer Sciences and Engineering, 2(11):* 50-54.

[17] Hahn, U. and Mani, I. 2000. The challenges of automatic summarization. *Computer, 33(11),* pp.29-36.

[18] i-net software. JOrtho - a Java spell-checking library. http://jortho.sourceforge.net.

[19] Jha, R.K. 2015. *NLP Driven Models for Automatically Generating Survey Articles for Scientific Topics* (Doctoral dissertation, The University of Michigan).

[20] Khan, A., Salim, N., and Kumar, Y.J. 2015. A framework for multi-document abstractive summarization based on semantic role labelling. *Applied Soft Computing, 30*, pp.737-747.

[21] Klein, D. and Manning, C.D. 2003. Accurate unlexicalized parsing. *In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pp. 423-430. Association for Computational Linguistics.

[22] Lin, C.Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop (Vol. 8)*.

[23] Martinez, D., Otegi, A., Soroa, A., and Agirre, E. 2014. Improving search over Electronic Health Records using UMLS-based query expansion through random walks. *Journal of biomedical informatics, 51*, pp.100-106.

[24] Mihalcea, R., Corley, C., and Strapparava, C., 2006. Corpus-based and knowledge-based measures of text semantic similarity. *In AAAI(6)*, pp. 775-780.

[25] Mihalcea, R. and Tarau, P. 2004. TextRank: Bringing order into texts. Association for Computational Linguistics.

[26] N.L.P.R. Group. 2010. Departments and Services. http://nlp.shef.ac.uk/.

[27] Porter, M. 2006. Porter Stemming Algorithm. http://tartarus.org/martin/PorterStemmer.

[28] Silber, H.G. and McCoy, K.F. 2002. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics, 28(4)*, pp.487-496.

[29] Spell, B. 2009. Java API for WordNet Searching (JAWS). http://lyle.smu.edu/ tspell/jaws/index.html.

[30] Wan, X., Cao, Z., Wei, F., Li, S., and Zhou, M. 2015. Multi-Document Summarization via Discriminative Summary Reranking. *arXiv preprint arXiv:1507.02062*.

[31] Wan, X., Yang, J., and Xiao, J. 2006. Incorporating cross-document relationships between sentences for single document summarizations. *In International Conference on Theory and Practice of Digital Libraries*, pp. 403-414. Springer Berlin Heidelberg.

[32] Wang, D., Zhu, S., and Li, T. 2013. SumView: A Web-based engine for summarizing product reviews and customer opinions. *Expert Systems with Applications, 40(1)*, pp.27-33.

[33] Wei, T., Lu, Y., Chang, H., Zhou, Q., and Bao, X. 2015. A semantic approach for text clustering using WordNet and lexical chains. *Expert Systems with Applications, 42(4)*, pp.2264-2275.

[34] WordNet. 2010. About Wordnet. Princeton University. http://wordnet.princeton.edu.