JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

SEMANTIC TEXT SUMMARIZATION

[1]Syed Ufaq Chashoo, [2]Kamal Kumar
[1]Mtech CSE, [2]HOD & Assistant Professor
[1] Kurukshetra University, Kurukshetra, India
[2]Swami Devi Dyal Institute of Engineering & Technology, Kurukshetra University, India

Abstract: Automatic Text Summarization is the process of generating automatic summaries from a document by extracting its principal statements and preserving the overall semantics of the text at the same time. Text Summarization is a potential solution to the problem of information overload the world is facing at the moment. Numerous text summarizers already exist in the literature, but they fail to preserve the semantics of the text. This research uses this semantic feature as a fundamental technique to find high quality summaries of documents. We use the distributional Semantic Model called word2Vec and compare the results with other state-of-the-art summarizers. We evaluated our results using ROUGE on DUC 2007 dataset. This system outperformed all other reference summarizers proving that using semantics as a feature increases the efficiency of the system.

***IndexTerms* –** Text Summarization, Distributional Semantic Model, word2Vec, Semantic Analysis

1. INTRODUCTION

Text analytics aids in extracting information from text. Text summarization produces short summaries and hence is a critical task in the Text analytics area. Text summarization is the process by which text is condensed for immediate consumption. It is used on many levels for proper understanding and efficient utilization of lengthy textual documents. It is the task of extracting imperative sentences from the document to generate automatic text summaries by enduring the overall semantics of the text. Thus, short summaries provide an abstract of the document, which is easy to comprehend for a user.

The task of Text summarization is efficient in many ways viz:

i. Reduces the reading time of the researcher during the selection of related articles.

ii. Various companies can use automatic Text summarization to effectively produce summaries in less time for decision making, financial Research and stock market information.

iii. It can be beneficial for e-learning systems to produce condensed reports related to a particular subject or topic.

iv. It can aid in sentiment analysis by providing summaries of both the positive and negative posts.

Abstractive, Extractive, and Hybrid are the three categories of summaries deduced from text. Sentence extraction is employed in extractive summarization [1][2][3]. The process of summarization begins with the extraction of sentences. The chosen subset is the document's significant set of sentences and, thus, the summary of the text. The statistical features like the position of a sentence, frequency of phrases or words, and subject phrases that indicate significant sentences in the text are employed to select the subset. The re-generation of retrieved information generates the abstractive Summaries. These summaries are premised on statistical features like the position of a sentence, frequency of phrases or words, subject phrases arduous rules [4]. The limitation of it is that it is highly dependent on the stylistic features and arduous rules and completely overlooks the semantics of the text. Hybrid summarization involves the combination of Extractive and abstractive summaries.

Automatic summarizers employ two learning schemes: supervised and unsupervised to generate automatic text summaries. The supervised learning approach produces satisfactory results but needs a massive amount of labelled training data, which is tedious and arduous to get. In contrast, unsupervised algorithms make use of linguistic and statistical aspects to obtain a summary. The linguistic and statistical features are premised on the specific intrinsic features of the text document like the position of a sentence, frequency of phrases or words, subject terms and sentence length, but ignores the critical and principal features of text data.

The main drawback of existing methods is that they concentrate on statistical attributes and ignore the text's semantics. Traditionally, these automatic summarizers are built on the assumption that statistical features are central to the summarization task and thus they miss an essential attribute of the text that is its meaning. Text summarization is the process by which text is condensed for immediate consumption. Text summarization is used for centuries, but automated text summarization was introduced in the early '90s. Since then, it has evolved in multiple forms. It uses various methods and means. These methods include extractive and abstractive summarization. In literature, text summaries lack an essential aspect used in human communication: text semantics. This Research aims to use text semantics as a feature to improve text summaries. Our system will use text semantics at the fine-grained level and then use the semantic features and the other surface features to produce better text summaries. We aim to enhance the system accuracy and evaluate the system against the existing baselines to prove our research goal.

In this System, we combine the two approaches, i.e., supervised and unsupervised approaches to produce the hybrid abstractive summarizer. Our system has 4 steps:1) Preprocessing to remove noise and inconsistencies, 2) to capture semantics by use the semantic distributional models viz Word2Vec, 3) to use vectors generated by each model for transforming sentences into separate three big-vectors of highly semantic extension, 4) clustering premised on semantic coherence and 5) Ranking algorithms rank sentences in each cluster to produce summaries by picking top sentences from each cluster. The remainder of the synopsis is structured into the consecutive sections: Section 2 discusses the Literature survey. Section 3 discusses the methodology employed. Section 4 presents results obtained, Section 5 presents conclusions and future work followed by references.

2. LITERATURE SURVEY

This portion covers the various aspects and methodologies employed in text summarization by artificial text summarizers and their relative pros and cons. This in-depth review of feature identification methods, techniques, and applications aims to draw conclusions and identify unresolved issues.

2.1. Supervised Learning Techniques

As a result of the recent development in machine learning techniques, machine learning methods generate extractive summaries. These methods employ manually selected significant features for training supervised learning methods to build a model. It is then utilized to classify sentences to arbitrate whether or not the sentence should be included in the summary. Naive Bayes and Support Vector Machines (SVM) models are employed with the derived features like content, event, relevance and topic, [5] to generate extractive summaries. [6] built an extractive summarizer with extracted linguistic and statistical attributes from the source document by employing machine learning methods. The sorting of sentences in the document could enhance the extractive summary results [7].

2.2. Unsupervised Techniques

Luhn took an excellent initiative in this field at the beginning of the 1950s. Luhn demonstrated that the importance of words in a document is directly proportional to their frequency. He also concluded that the most frequent terms are either descriptive or topic terms. Thus, the sentences bearing these more significant terms must be appended in summary [8]. [9] incorporated more features in Luhn’s work that enhance the score of significant sentences for text summarizing. The groundwork of abstract summarizers was set by [10] and proposed the concept of language generators. They also address that summaries could be generated by incorporating the sentences in the summaries that are not included in the document’s text. [11] use ranking algorithms for sentence scoring of a single document and employs the statistical approaches suggested by Edmundson. To obtain a better summary [12] used Latent Semantic Analysis to determine the coherence between the sentences and topics. TextRank proposed by [13] to generate extractive summaries of documents using the cumulative weighted correlation among phrases.

2.3. Query-based Technique In this technique, the document's text is scanned for the query keywords of the user. The sentences with query phrases are assigned high ranks than sentences by single query words. The higher-scoring sentences incorporated with other integral components are the summary's output. The outcome of these summarizers is the fusion of all the extracts [14]. [15] employed two methods: a query expansion approach and graph basis summarization to elucidate limited content in the original queries. Instead of using external resources like WordNet[16], they employed the document set on which the summarization is done to expand queries .SVR (Support Vector Regression) is employed by [17] to determine the relevant sentences in a document for summary as output for the user query. [18] employed MMR (Maximal Marginal Relevance) to combine relevant query and novel content for the text summarizing task.

2.4. Graph and Network Techniques [19] introduced graph-based approaches construct extractive single-document summaries by employing supervised and unsupervised learning schemes. The objective was to determine the relevant sentences by extracting statistical attributes using supervised and unsupervised methods and to analyze the coherence among sentences using graph approach. [20] computed the similarity adjacency networks of words to determine authorship. They identified the author by using text as a graph. For summarization of multiple documents, [21] use multi-layer graph methods where nodes represent sentences and edges represent the relation between the two sentences.

2.5. Neural Network Techniques With the development in deep learning techniques and lower costs of memory, these techniques have gained a lot of popularity. The neural-network summarizers achieve better performance with the least human intervention if adequate training data is acquired than traditional automatic summarizers [22]. [23] employed continuous vectors based on a neural network to generate extractive summary and computed better results and thus laid the ground to utilize neural networks for text summarization.[24] are the first to propose an abstractive summarizer that generates text summaries by using CNN (Convolutional Neural Networks). [25] built an abstractive summarizer using CNN and other Neural Networks to extend the work presented by [24]. [26] employed a RNN (Recurrent Neural Networks) with attentional encoder decoder to generate an abstractive summary. [27] presented COPYNET, a sequence-to-sequence learning algorithm that copies text segments in source document at particular time intervals. [28] used neural networks with a pointer-generator technique to respond to the off-vocabulary issue. [29] build a large comprehensive Chinese corpus for the summary generation. [30] presented neural network premised on distraction technique that permits to distract among various different segments of the input text. [31] presented a neural network method based on semantic relevance that generates semantically significant summaries of Chinese data.[32] employs a bidirectional LSTM encoder to generate an extractive summary.

[39] capture semantics of text are and preserved it as a crucial component for summarization. The system captures semantics and produces high-quality summaries using the distributional semantic model Word2Vec. All the sentences are represented in their semantic extensions using word2vec by forming big-vectors. ROUGE is employed to assess the performance of a summarizer on the DUC-2007, and the results are compared with four existing baselines. The system generates summaries of the length of 25% and 50% of the original document.

The main limitation of the machine learning-based and neural network approaches outlined above is that they require massive data and effort to generate high-quality summaries. Moreover, the existing automatic summarization methods concentrate on statistical attributes and ignore the text's semantics. Traditionally, these automatic summarizers are built on the assumption that statistical features are central to the summarization task, and miss an essential attribute of the text that is its meaning. This Research aims to use text semantics as a feature to improve text summaries and will use text semantics at the fine-grained level and then use the semantic features and the other surface features to produce better text summaries. We aim to enhance the accuracy and evaluate the system against the existing baselines to prove our research goal.

3. METHODOLOGY

This section describes the system for text summarization to produce the hybrid abstractive summarizer by utilizing both supervised and unsupervised approaches. Our system will use text semantics at the fine-

grained level and then use the semantic features and the other surface features to produce better text summaries. Our system has 4 steps:
1) Preprocessing to remove noise and inconsistencies,
2) to capture semantics by the use of the semantic distributional model viz Word2Vec
3) to use vectors generated by the model for transforming sentences into separate bigvectors of highly semantic extension,
4) clustering premised on semantic coherence and
5) Ranking algorithms rank sentences in each cluster to produce summaries by picking top sentences from each cluster.

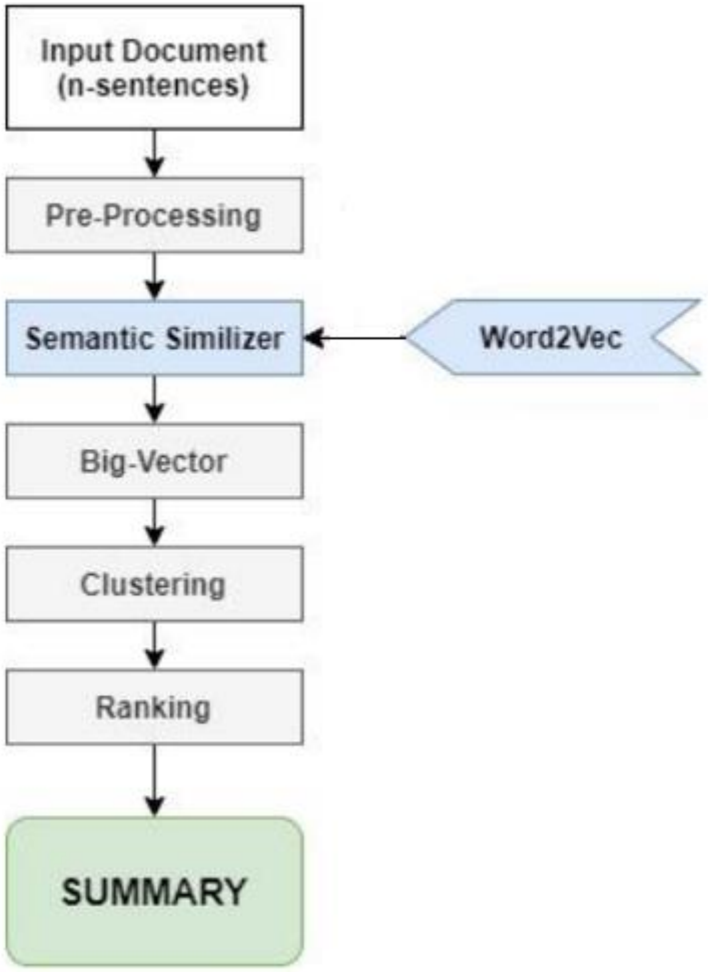


Fig. 1 Overall working of the model

3.1. Preprocessing

Preprocessing is the primary and first step in our classification process. It cleans data for uniformity, removes noise and inconsistencies. Our Preprocessing algorithm has the following steps:
i. Removing URL's: URL's are meaningless and, thus removed to reduce noise from the main text.
ii. Tokenization: Each sentence is split into words (tokens). We use Stanford Core NLP [33] for this process.
iii. Removing punctuation and numbers: Punctuations and numbers do not add any meaning during the classification task and hence removed.
iv. Removing Stop-words: Stopwords are meaningless and do not emphasize any sentiment and, are thus, removed to reduce noise from the main text.

v. Lowercase: For the sake of uniformity, the text is converted to lower case. vi. Lemmatization: Next, lemmatization is applied to words to reduce them to their stems. It is done using Stanford core NLP package [33].

3.2. Procuring semantics by employing various Semantic Distributional Models:

For capturing the meaning of text, we employed semantic distributional models. Since distributional semantic algorithms are trained on large scale datasets and are thus generic and used on different domains, these algorithms do not require any lexical and linguistic tuning and hence act as excellent semantic models. These algorithms are independent of other sources for procuring semantic extensions of text. The distributional models are premised on the distributional hypothesis, which states that words used in the same context have similar meanings. The following t models are employed to capture the semantic similarity of text as text semantics are valuable and applicable in different methods like [35][36]. i. Word2Vec: Word2Vec[34], a two-layer neural network, processes textual data and produces vectors for each given word. Word2Vec uses a neural network architecture to transform input text into a series of word tokens and produces vectors as output after processing. These generated vectors are rich semantically and act as feature vectors. By utilizing these two layers of the neural network and series of computation, the algorithm converts data into the format popularly known as vector space dimensional model. Word2Vec has two architectures: CBOW Continuous bag-of-words () and Skip-gram model. CBOW predicts the current word from its context and skip-gram predicts the context given the word.

3.3. Big- Vectors Formation We use word vectors generated by the model for transforming sentences into separate big-vectors [36]. All the tokens of the sentence are fed to the three semantic model viz Word2Vec. Then, the retrieved word vectors from each model are concatenated to generate their respective big-vector. Thus, the three different big-vectors are used for transforming each sentence in the text into its highly semantic extension.

3.4. Clustering

We will use a clustering algorithm to group each semantically coherent retrieved big-vectors into primary clusters. K-means [37] clustering is used for clustering premised on semantic coherence.

3.5 Ranking Algorithm

The extracted summary is generated using our new ranking algorithm, by ranking the sentences in each cluster. The ranking algorithm employs different statistical attributes like cumulative cosine coherence, Cue phrases, length of sentence, noun and verb phrases, sentence position, Tf_Idf and Proper Noun for extraction of top n sentences from each semantically rich cluster.) Ranking algorithm rank sentences in each cluster to produce summaries by picking top sentences from each cluster, normalizes them, and then sums these scores to get the normalized score.

4. EXPERIMENTAL SETUP AND RESULTS

4.1 Dataset

We use the primary dataset acquired from the ACQUAINT corpus from the DUC-2007i (Document Understanding Conference) [38]. The DUC is a set of summarizing activities, and since 2001, the NIST (National Institute of Standards and Technology) has been working on a colossal text corpus to evaluate automatic summarizers. This dataset contains News articles from the Xinhua News Agency and the New York Associated Press. There are fortyfive separate topics in it, and each topic has significant 45 documents linked with it. The NIST has developed four principal summaries of roughly 250 words for each issue that serve as a baseline for other text summarizers

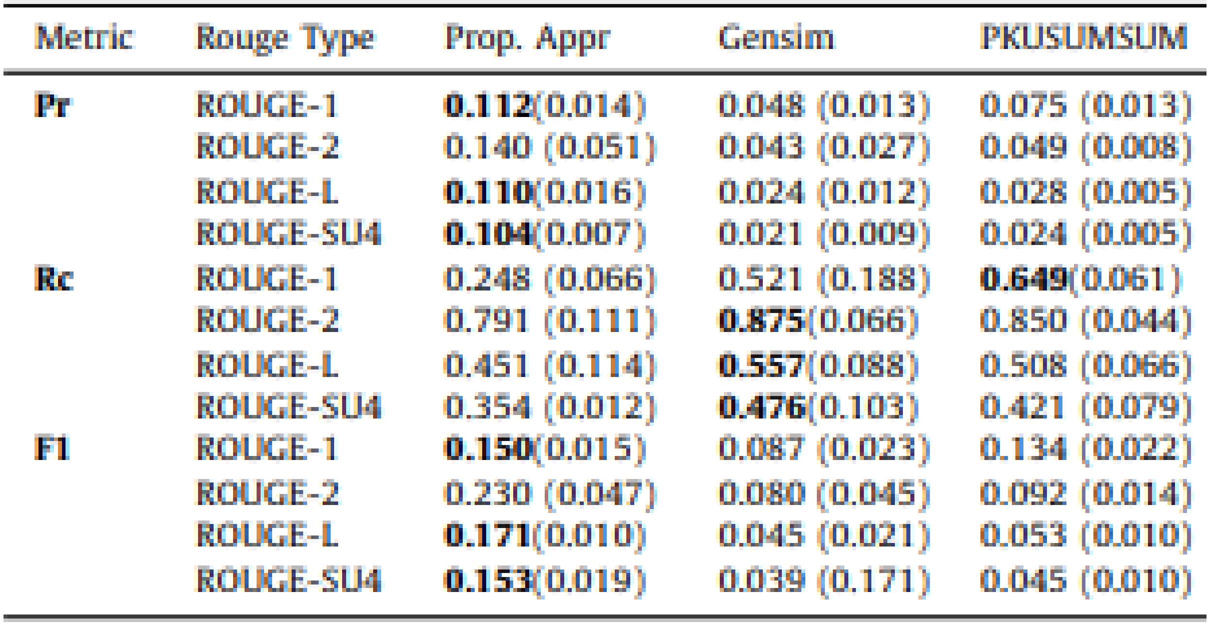
4.2 Baselines

We evaluated our results against the following state-of-the art existing systems given as:

GENSIM summarizer is an implementation of the TextRank algorithm TextRank is a graph-based ranking system that calculates the importance of a sentence in the text, presented as a vertex, iteratively from the graph's global state. If a vertex is linked to another vertex, the linking vertex receives one vote, increasing the vertex's connectivity to other vertices and hence its rank.

PKUSUMSUM is a Java summarization framework with ten summarization algorithms and support for several languages. It also supports three summarization tasks viz: Single-document summarization, multi-document summarization, and Topic-based multi-document summarization. It includes reliable and diverse summarization approaches, and its performance is sufficient to serve as a benchmark for our review. It provides various summarizing methods such as, Centroid, LexPageRank, and TextRank. We employed a single-document summarizer with the LexPageRank algorithm for summarization in our evaluation These systems use various summarizing techniques, therefore comparing our system's performance to theirs leads to a full and extensive comparison. Furthermore, because these systems are open source, experiments can be readily reproduced.

Average Summarization result of 50% summary length



Metric	Rouge Type	Prop. Appr	Gensim	PKUSUMSUM
Pr	ROUGE-1	**0.112**(0.014)	0.048 (0.013)	0.075 (0.013)
	ROUGE-2	0.140 (0.051)	0.043 (0.027)	0.049 (0.008)
	ROUGE-L	**0.110**(0.016)	0.024 (0.012)	0.028 (0.005)
	ROUGE-SU4	**0.104**(0.007)	0.021 (0.009)	0.024 (0.005)
Rc	ROUGE-1	0.248 (0.066)	0.521 (0.188)	**0.649**(0.061)
	ROUGE-2	0.791 (0.111)	**0.875**(0.066)	0.850 (0.044)
	ROUGE-L	0.451 (0.114)	**0.557**(0.088)	0.508 (0.066)
	ROUGE-SU4	0.354 (0.012)	**0.476**(0.103)	0.421 (0.079)
F1	ROUGE-1	**0.150**(0.015)	0.087 (0.023)	0.134 (0.022)
	ROUGE-2	0.230 (0.047)	0.080 (0.045)	0.092 (0.014)
	ROUGE-L	**0.171**(0.010)	0.045 (0.021)	0.053 (0.010)
	ROUGE-SU4	**0.153**(0.019)	0.039 (0.171)	0.045 (0.010)

As evident from the results of the summarization experiment, our proposed system performs better than the state-of-art baselines and confirm the competitive efficiency of the proposed algorithm

As expected, recall values rise as summary length increases, but still remain lower than the PKUSUMSUM.. Thus, we conclude usage of semantic features allows our system to generate better summaries. As far as the evaluation of the summaries is concerned, the macro-average of the F-scores of different ROUGE metrics is 18.25%, while as those for the other baselines is 6% for Gensim, 15% for PKUSUMSUM for 25% summary length.

The use of semantics as a feature for text summarising system is attributed to higher F-scores of our system, and thus we conclude that the system's performance for producing summaries increases with the use of semantics. Furthermore, as compared to other baselines, we obtained higher precision and F-scores during the system evaluation. Our approach has a poor recall since it rejects some sentences that are statically different but semantically similar.

5. CONCLUSIONS AND FUTURE SCOPE

The research proposes a text summarization technique based on the distributional hypothesis for capturing the semantics of the text in order to produce better summaries using text summarization. Our proposed technique outperforms the baselines in terms of appropriateness, dependability, and throughput, as determined by evaluation and comparative analysis. Our main conclusions are:

(1) capturing semantics and using it as a feature for summarization helps to improve the precision of our summaries.

(2) combining semantic features along with other features tend to produce consistently good summaries.

(3) usage of distributional semantic hypothesis tends to produce good results in summarization work as well. The primary disadvantage of the proposed system is that using distributional semantic model is computationally expensive and time consuming.

Our future research will deal with (1) using more than one distributional semantic models for capturing semantics in the text so that semantics are captured at the fine grain level;

(2) Improvement of ranking algorithms by exploring more semantic features to be incorporated into our ranking algorithms, as semantic features tend to improve overall system performance;
(3) testing the technique on more than one dataset.

REFERENCES

[1].Rotem, N. "Open text summarizer (ots)." Retrieved July 3, no. 2006 (2003): 2006.

[2].Lloret, Elena, María Teresa Romá-Ferri, and Manuel Palomar. "COMPENDIUM: A text summarization system for generating abstracts of research papers." Data & Knowledge Engineering 88 (2013): 164-175.

[3].Sevilla, Antonio FG, Alberto Fernández-Isabel, and Alberto Díaz. "Grafeno: Semantic graph extraction and operation." In 2016 eleventh international conference on digital information management (icdim), pp. 133-138. IEEE, 2016.

[4].Barros, Cristina, Elena Lloret, Estela Saquete, and Borja NavarroColorado. "NATSUM: Narrative abstractive summarization through crossdocument timeline generation." Information Processing & Management 56, no. 5 (2019): 1775-1793.

[5].Wong, Kam-Fai, Mingli Wu, and Wenjie Li. "Extractive summarization using supervised and semi-supervised learning." In Proceedings of the 22nd international conference on computational linguistics (Coling 2008), pp. 985-992. 2008.

[6].Neto, Joel Larocca, Alex A. Freitas, and Celso AA Kaestner. "Automatic text summarization using a machine learning approach." In Brazilian symposium on artificial intelligence, pp. 205-215. Springer, Berlin, Heidelberg, 2002.

[7].Conroy, John M., and Dianne P. O'leary. "Text summarization via hidden markov models." In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 406-407. 2001.

[8].Luhn, Hans Peter. "The automatic creation of literature abstracts." IBM Journal of Research and development 2, no. 2 (1958): 159-165.

[9].Edmundson, Harold P., and Ronald E. Wyllys. "Automatic abstracting and indexing—survey and recommendations." Communications of the ACM 4, no. 5 (1961): 226-234.

[10]. Baxendale, Phyllis B. "Machine-made index for technical literature—an experiment." IBM Journal of Research and development 2, no. 4 (1958): 354-361.

[11]. Afantenos, Stergos, Vangelis Karkaletsis, and Panagiotis Stamatopoulos. "Summarization from medical documents: a survey." Artificial intelligence in medicine 33, no. 2 (2005): 157-177.

[12]. Gong, Yihong, and Xin Liu. "Generic text summarization using relevance measure and latent semantic analysis." In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 19-25. 2001.

[13]. Mihalcea, Rada, and Paul Tarau. "Textrank: Bringing order into text." In Proceedings of the 2004 conference on empirical methods in natural language processing, pp. 404-411. 2004.

[14]. Pembe, F. Canan, and Tunga Güngör. "Automated querybiased and structure-preserving text summarization on web documents." In Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications, İstanbul. 2007.

[15]. Zhao, Lin, Lide Wu, and Xuanjing Huang. "Using query expansion in graph-based approach for query-focused multi-document summarization." Information processing & management 45, no. 1 (2009): 35-41.

[16]. Li, Wenjie, Wei Li, Baoli Li, Qing Chen, and Mingli Wu. "The hong kong polytechnic university at duc 2005." In Proceedings of Document Understanding Conferences. 2005.

[17]. Ouyang, You, Wenjie Li, Sujian Li, and Qin Lu. "Applying regression models to query-focused multi-document summarization." Information Processing & Management 47, no. 2 (2011): 227-237.

[18]. Carbonell, Jaime, and Jade Goldstein. "The use of MMR, diversitybased reranking for reordering documents and producing summaries." In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 335-336. 1998.

[19]. Mao, X., H. Yang, Shaobin Huang, Y. Liu and Rongsheng Li. “Extractive summarization using supervised and unsupervised learning.” Expert Syst. Appl. 133 (2019): 173-181.

[20]. Amancio, Diego R., Filipi N. Silva, and Luciano da F. Costa. "Concentric network symmetry grasps authors' styles in word adjacency networks." EPL (Europhysics Letters) 110, no. 6 (2015): 68001.

[21]. Tohalino, Jorge V., and Diego R. Amancio. "Extractive multidocument summarization using multilayer networks." Physica A: Statistical Mechanics and its Applications 503 (2018): 526-539.

[22]. Dong, Yue. "A survey on neural network-based summarization methods." arXiv preprint arXiv:1804.04589 (2018)

. [23]. Kågebäck, Mikael, Olof Mogren, Nina Tahmasebi, and Devdatt Dubhashi. "Extractive summarization using continuous vector space models." In Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC), pp. 31-39. 2014.

[24]. Rush, Alexander M., Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization." arXiv preprint arXiv:1509.00685 (2015).

[25]. Chopra, Sumit, Michael Auli, and Alexander M. Rush. "Abstractive sentence summarization with attentive recurrent neural networks." In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 93-98. 2016.

[26]. Nallapati, Ramesh, Bowen Zhou, Caglar Gulcehre, and Bing Xiang. "Abstractive text summarization using sequence-to-sequence rnns and beyond." arXiv preprint arXiv:1602.06023 (2016).

[27]. Gu, Jiatao, Zhengdong Lu, Hang Li, and Victor OK Li. "Incorporating copying mechanism in sequence-to-sequence learning." arXiv preprint arXiv:1603.06393 (2016).

[28]. See, Abigail, Peter J. Liu, and Christopher D. Manning. "Get to the point: Summarization with pointer-generator networks." arXiv preprint arXiv:1704.04368 (2017).

[29]. Hu, Baotian, Qingcai Chen, and Fangze Zhu. "Lcsts: A large scale chinese short text summarization dataset." arXiv preprint arXiv:1506.05865 (2015).

[30]. Chen, Qian, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. "Distraction-based neural networks for document summarization." arXiv preprint arXiv:1610.08462 (2016).

[31]. Ma, Shuming, Xu Sun, Wei Li, Sujian Li, Wenjie Li, and Xuancheng Ren. "Query and output: Generating words by querying distributed word representations for paraphrase generation." arXiv preprint arXiv:1803.01465 (2018).

[32]. Paulus, Romain, Caiming Xiong, and Richard Socher. "A deep reinforced model for abstractive summarization." arXiv preprint arXiv:1705.04304 (2017).

[33]. Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014, June). The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations (pp. 55- 60).

[34]. Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[35]. Canales, Lea, Carlo Strapparava, Ester Boldrini, and Patricio Martinez-Barco. "Intensional learning to efficiently build up automatically annotated emotion corpora." IEEE Transactions on Affective Computing 11, no. 2 (2017): 335-347.

[36]. Jan, R., & Khan, A. A. (2020). Emotion mining using semantic similarity. In Natural Language Processing: Concepts, Methodologies, Tools, and Applications (pp. 1115-1138). IGI Global.

[37]. Hartigan, John A., and Manchek A. Wong. "AK-means clustering algorithm." Journal of the Royal Statistical Society: Series C (Applied Statistics) 28, no. 1 (1979): 100-108.

[38]. Vorhees, Ellen, and David Graff. AQUAINT-2 Information-retrieval text: Research collection. Linguistic Data Consortium, 2008

[39]. Mohd, Mudasir, Rafiya Jan, and Muzaffar Shah. "Text document summarization using word embedding." Expert Systems with Applications 143 (2020): 112958.