Journal of Theoretical and Applied Information Technology
30th September 2023. Vol.101. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

UNSUPERVISED LEARNING ONTOLOGY BASE TEXT SUMMARIZATIONS APPROACH WITH CELLULAR LEARNING AUTOMATA

ELHAM GHASEMI[1], VAHID RAFIEI[2], GOLSHID RANJBARAN[3]

[1]Department of Computer, Science and Research Branch, Islamic Azad University, Tehran, Iran
[2]Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX, USA
[3]Department of Computer, Science and Research Branch, Islamic Azad University, Tehran, Iran
E-mail: [1]elham_ghasemi@srbiau.ac.ir, [2] vahid.rafiei@utdallas.edu, [3]golshid.ranjbaran@srbiau.ac.ir

ABSTRACT

Text summarization is the procedure for generating a short copy of a given text. The main objective of text summarization is to create an outline that encompasses the text's main content. In this paper, a new model based on ontology, unsupervised learning, and cellular learning automata is proposed for the text summarization task. For this purpose, using the ontology, concepts of sentences are extracted and mapped to some clusters of sentences with similar meaning, where the most appropriate sentence is selected for the summarization task. The clustering has been done by using K-means unsupervised learning on a corpus of English sentences. Cellular Learning Automata (CLA) is applied for the calculation of *n*-grams and extracting the summary content. The results were evaluated using the ROUGE-2 method and showed that the quality of the summary text improved by an average of 19.26% compared to other works in the literature.

Keywords: *Text Summarizations, Unsupervised Learning, CLA, Ontology*

1- INTRODUCTION

Summaries are a shortened form of text that give readers an overview of the content. Reading a summary allows the reader to make a decision about reading the full text. Extractive and Abstractive Summarization Systems are the methods to outline the content utilizing software programs [1-3]. Extractive outline approach is based on selection of representative content from the given content information. Abstractive approach is concerned with producing outline by following the sense and nature of the content. In abstractive method, we may utilize new words or sentences for detecting the extensive content while in extractive outline, we will choose characteristic set of words and sentences. Initial summary systems were created based on simple formal features such as the position of sentences in a document [4], the frequency of words [5, 6], or the existence of words and expressions of reference [6]. A number of more advanced methods used machine learning techniques to determine the best set of features for extraction [7]. The principle issue for creating an extractive programmed content summarization is to distinguish the most important data in the source document. Although, a few methodologies guarantee being language independent, most of them utilize some level of language learning like lexical rules [8], key-phrases [9] or characteristic examples for supervised learning [10, 11], as in LSTM-CNN deep learning approach [12], clustering and optimization [13] and improved semantic graph approaches [14].

The ontology can be considered as a set of entities and the relationship between them. The ontology, together with the set of individual instances, forms the knowledge classes, which represent the concepts and are the most important part of ontology. An ontology defines a set of vocabulary and includes terms, which help to remove any ambiguity from the text [15]. In [16], an ontology-based approach for creating a customizable summary was created which can be identified during the survey summary and preferences. RDF diagrams were constructed and appropriate RDF sentence scores were obtained for user-significant sentences. In [17], an abstract summary was presented which used Yago's ontology with entity recognition and disambiguation, sentence ranking and sentence selection. In [18], an ontology called as Texminer was used to extract important sentences with rhetorical structure. The ontology helped to incorporate all the important concepts of the topics, and to obtain semantic information of the concepts to facilitate abstraction.

In graph-based methods, combining sentences helps to remove the extra parts of a sentence. It is assumed that there are many similar sentences in the text, and this similarity helps to merge sentences in summarization process. In [19], a combination of phrases and unsupervised methods were used to produce summaries of natural language. The graph structure was also used to find similar structural sentences, and then the paragraph text was generated. In [20], based on grammar correction, information-based summation was created by finding the shortest path in word-graph, but due to information-based ranking, his work suffers from lack of linguistic quality. In [21], research of [20] was developed with the help of inference graph creation, additional information was eliminated by word graphs, so that important sentences were extracted from the text and could produce abstract summaries. They used WordNet to find relationships between words and applied this information to merge graph nodes. Information content and grammatical issues were selected as factors for deciding the best paths to the graph, and to generalize summary. In [22], "word-graph" fragment graphs were used to reduce the graph size. Document graph representation is performed by using the similarity relationship between sentences, where the similarity between two sentences is used as the weight of the edge between two nodes.

In cluster-based summarization, the narrative text classification is used as an automatic keyword extraction. Three popular methods are TFIDF, KEA, and Keyterm, which are used to extract keywords from the content validity of the pages. They employ ANOVA tests to analyze and rank data with acceptable percentages and quality. The evaluation tests show that the key phrases extracted from the narrative text are significantly better than those obtained from the full text in the pages. This shows that the classification of narrative text to extract key terms is essential for efficient document summarization [23]. In [24], authors presented a cluster-based approach that consisted of two steps, the sentences being first clustered and then the representative sentences based on each cluster were defined and extracted, and a discrete differential evolution algorithm was used to optimize the objective functions. The results was evaluated by the ROUGE-1, ROUGE-2 and ROUGE-su4metrics methods.

Other methods of summarization include the use of graphs and neural network models for extracting sentences from the text. These methods recognize the most important sentences using different linguistic properties. They consider the correlation between textual units to eliminate weak connecting sentences in the summary. To calculate the similarity degree between sentences, a proper indexing weight is assigned according to the document conditions. The process is as follows: a) Pre-processing: Parsing the document and generating sentences, b) Graph structure: Each sentence is considered as a node where each node has its properties that specifies its behavior, c) Sentence ranking: The relationship between sentences is calculated based on collected penalties and rewards [25]. In this way, after extracting their important properties, the sentences are linearly combined to reveal the significance of each sentence. Two algorithms that are used for measure similarity are bee colonies and cellular learning automata [26].

Research Objectives are developing an unsupervised learning approach for text summarization that leverages the hierarchical structure and semantic relationships encoded in an ontology and conduct a comparative analysis with other text summarization techniques, including supervised and semi-supervised methods, to highlight the advantages and limitations of the proposed approach.

Most text summarization methods come with important shortcomings, the consequence of which is to ignore the semantic relationships of phrases, synonyms, and words that have the same pronunciation. In this study, we propose a method for extraction summarizing of texts trying to overcome abovementioned shortcomings. In order to improve the accuracy of post-processing summary, we use a graph-ontology method as well as unsupervised learning for text corpus to increase understanding of concepts. Finally, we use cellular learning automata model for text summarization.

2- METHODOLOGY

The steps of proposed algorithm for summarization are as follows:

1) Pre-processing and disambiguation
2) Take each document as a graph and break the text into sentences.
3) Determine the importance of each sentence by ontology and unsupervised K-means clustering.
4) Compute *n*-grams with CLA to increase accuracy
5) Determine the closest sentence (most factor sentence) for each cluster center to summarize and determine the degree of the importance of the sentence in the final summary.

The general approach is shown in Figure 1:

In the proposed approach, we expect that due to the clustering of concepts in the ontology section as well as the addition of CLA, we get more accurate summarization results.

2-1- Pre-Processing and Disambiguation

The text is broken into sentences. Stop-words and high vocabulary and high frequency phrases are excluded from the text because they are not useful in recognizing the relevance of sentences. The output of this step is a matrix whose elements at each row are important and processable words of a text sentence. Tokenization is a procedure for breaking a content stream in words, sentences, phrases and other significant items called tokens. Vectorization changes over these content records to a network of tokens dependent on their frequency in the text.

2-2- Document (corpora) to graph conversion with ontology

Each document is considered a graph. The vertices are the concepts embedded in the document that are related to the domain ontology, and the edges represent the relationship between these concepts. For each sentence in the text, its containing concepts are linked to their related concepts in other sentences, thus expanding the structure of the graph [27]. There are two ways to determine the path value from any concept to another, i.e., the degree of association between two concepts: If these two concepts are directly related, we can use their predefined relationship degree. If the two concepts are not directly related, then we must determine the amount of implicit communication between them in a proper way. For this purpose, regardless of the type of relationship between intermediary concepts, the minimum weight of the relationship between the concepts or concepts examined between the two concepts is referred to as the weight of the relationship between the two concepts [28]. Most existing methods of indexing and weighting the terms and concepts contained in documents are based on the presence or absence of a word in a set of documents. In general, the index words describe the content of a document in different dimensions. In indexing algorithms, each of the index terms is weighted based on its importance in the document. So far, various weighing methods have been presented and tested [29–31].

To apply ontology to corpora, it is first and foremost useful to define a preliminary conceptual structure for the domain from introductory or didactic texts on the studied domain. Secondly, we recommend performing a relationship extraction using the bottom-up method (based on the corpus): it consists of taking the list of extracted terms, searching for each term in its context, and studying its matches. In order to observe the conceptual relationships that each of them has with the other terms. Finally, we recommend applying the top-down method, which consists of looking for conceptual information in lexicographic works [32].

2-3- Unsupervised Learning for Graph Ontology

The unsupervised learning algorithm is used to detect a group of sentences with similar meaning to choose the best sentence representative of each group for summarization. In particular, we used the well-known K-means algorithm. In the K-means clustering, firstly we select K random members and consider them as the centers of the clusters. The remaining N-K members are then allocated to the nearest cluster. After allocation of all members, the cluster centers are recalculated and the members are reallocated to the clusters according to the new centers. This process will continue until the cluster centers remain constant. In other words, if we have data sets of N data points $(x_1, \dots x_N)$ and number of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

K clusters $(C_1, \dots, C_K), K \leq N$, the K-means goal is to minimize the mean squared of the similarity interval $\|x_j - c_i\|^2$ [33]:

$$\min \sum_{i=1}^{K} \sum_{x_j \in C_i} \|x_j - c_i\|^2 \quad (1)$$

where c_i is the center of cluster C_i. The reasons for the utility of the K-means method are its ease, simplicity, scalability, convergence speed, and consistency with sparse data.

The K-means algorithm provides an easy way to implement an approximate solution of the equation. The reasons for the utility of the K-means method are its ease, simplicity, scalability, convergence speed, and consistency with sparse data. The K-means clustering algorithm can be considered as a slope or descending gradient method. Cluster centers begin and regularly update cluster centers in order to reduce the objective function in the equation K-means always converges to the minimum location.

Implementation of the standard K-means involves continuous and sequential repetition. Each iteration of the entire data set visit is required to assign the data to the corresponding clusters. At the end of each iteration the new centers are calculated so that the next iteration uses the new centers. After a certain number of such duplicates the centers will remain the same [34]. The operation of the algorithm must be such that the boundary between the data set that is likely to be switched to another cluster and the data that holds and maintains the cluster that it owns during the next iteration is tracked. As the implementation of the K-means algorithm progresses, the centers get closer to their final position. As the number of iterations increases, the centers become less divergent from their current position and consequently fewer data items are controlled and reviewed [35].

In fact, most clustered data subjects or clusters whose centers move slowly should not be affected by motion. They will remain part of the same cluster at the next iteration and have fewer points of motion. The ability to detect which data topics affect motion means that we do not need to visit the entire dataset, and only a small part of the dataset is sufficient. Before deciding which data issues to include in the "demarcation," we need to determine the criteria that must be implemented by the data elements in a way that is capable of incorporating the "demarcation" element. Assume point p is part of cluster c (Figure 2). Point p is part of cluster C, and the distance between p and A is less than its distance to A and also less than its distance to B. We want to know how far point p is from jumping to another cluster. Obviously, the formula would be:

$$ep = \mathrm{Min}(d_{PA} - d_{PC}, d_{PB} - d_{PC}) \quad (2)$$

Where ep is the distance between p to the nearest boundary. It can be said that point p is about the size of ep for switching to another cluster. At the end of duplicate i, the centers need to be updated and updated according to the new order.

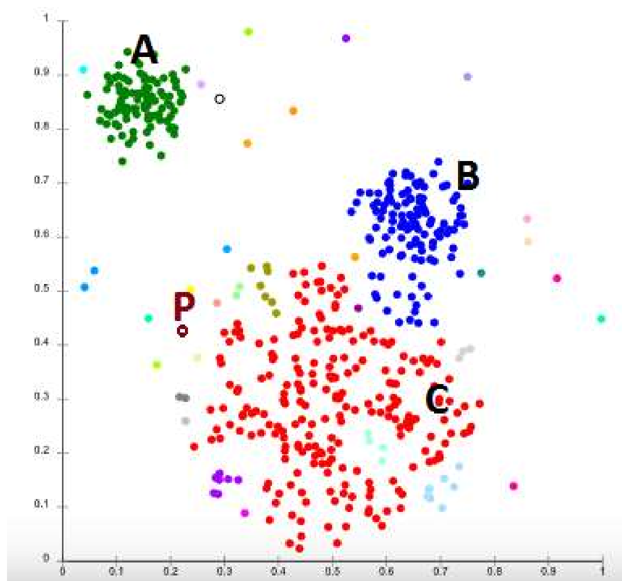

Figure 2; Three hypothetical clusters of data

Suppose center A has moved A and center B has moved B and C has moved C. The worst-case scenario for point p is that point C is further away from point p as it gets closer to point A and B, so what would be the conditions for P to get closer to cluster C Obviously it will be as follows:

$$ep > |CC| + |AA| \quad (3)$$

$$ep > |CC| + |BB| \quad (4)$$

To simplify the algorithm and reduce the computation we can combine terms (3) and (4):

$$ep > 2 * \mathrm{Max}(|AA|, |BB|, |CC|) \quad (5)$$

Which is generally a way to determine whether a point is part of a "demarcation" list or not [36], but because we check and control the differences and inequalities for the data elements in each iteration where we started Returns. To prevent such computations, we can adjust all data elements at wider intervals, based on the research of Dierckens et al. [37]. To evaluate the similarity between a cluster and a graph of equation (6), we use [38, 39]:

$$\mathrm{Similarity}(\mathrm{C_i}, \mathrm{O_j}) = \sum_{\mathrm{v_k}|\mathrm{v_k} \in \mathrm{O_j}} \mathrm{w_{k,j}} \quad (6)$$

$$w_{k,j} = 0, O_j v_k \in C_i$$

We used an unsupervised algorithm to discover groups of sentences with similar meaning in the corpus. Then we can choose the best sentence representative of each group to summarize. Figure 3 shows the clustering algorithm which is similar to approach of [37]. We use some criteria like elbow and silhouette method to determine the best number of clusters.

1. *Define constant WIDTH*
2. *Define intervals li = l_i * WIDTH(i+1) * WIDTH) and tag them with value i * WIDTH*
3. *Mark the entire data set to be visited*
4. *For each point to be visited:*
5. *Compute e = min(d_{pci} - d_{pcw}) where C_w is the center of the winner (closest) cluster and C_l, l=1..k,l≠w stands for all other centroids*
6. *Map all points with i * WIDTH <e < (i+1) * WIDTH to interval i * WIDTH where i is a positive integer*
7. *Compute new centroids Cj, where j=1..k and their maximum deviation D = max(|CjCj'|)*
8. *Update Ij's tag by subtracting 2 * D (points owned by this interval got closer to the edge by 2 * D)*
8. *Pick up all points inside intervals whose tag is less or equal to 0, and go to 4 to revisit them*

Figure 3: Concept clustering algorithm based on research by Dierckens et al. [39]

The community of Artificial Intelligence and Knowledge Engineering has proposed several definitions to identify the nature of computer ontology and ontology in general. From the early 90s, Gruber proposed the following definition: An ontology is a "formal knowledge representation" [40]. He then went on to specify that an ontology is a "formal and explicit specification of a shared conceptualization" [41]. Knowledge must therefore be not only formalized but also shared by several people. Guarino then reviews the definitions offered by Gruber, defining ontology as a "common and shared understanding of a domain that can be communicated between people and systems" [41]. The ontology must be understood by many people, but also understood by the software. Yoshioka recalls that the representation of ontology is a set of object classes [42]. Bachimont recalls that ontology refers to logic. He defines ontology as "a rigorous and structured vocabulary description, in the logical sense, a specialized field" [42].

Specifically, an ontology is a set of concepts structured in a network of dependencies that has hierarchical relationships (i.e., a property inheritance between concepts) and semantic relations (describing the properties of concepts or roles between concepts).

We classify the sentences into the nodes of a predefined hierarchical ontology, that is, a classification task. In addition, the reliability weight of the node computed by the classifier, enables us to identify the main issues of a document. The ontology we use is not domain specific. Also, the hierarchical classifier that maps sentences to nodes does not need tagged data in the training phase.

We compute the set of properties for each sentence based on the hierarchical classifier output. By collecting the nodes computed by the hierarchical classifier, we create a handbag for each sentence. If a sentence is mapped to several sub trees in the class, we will insert all nodes from each sub tree. By adding tag labels, we can create tags in a document:

$$\mathrm{w_d(t)} = \sum_{\mathrm{i} \in \mathrm{sentences\ (d)}} \mathrm{conf(t,i)} \quad (7)$$

where $w_d(t)$, is the weight of tag t and $conf(t,i)$ is the confidence value of the *t* in *i*, the highest tags can be interpreted as the main topics of a document [43].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2-4- Compute N-Grams With CLA

Cellular Automata (CA) consist of a regular network of cells, each of which is an automaton with a finite set of states like on and off. For each cell, a set of cells, called its neighbors, is defined which make interactions with that particular cell. Typically, the rules for updating the cell's status are similar for all the cells and will not change over the time [44]. A Learning Automata (LA) can be considered as an abstract object with a finite number of actions. The LA works by choosing an action from its set of actions at each time step. The action is evaluated by a randomized environment response and the automaton uses the environment reward or penalty to choose its next action. If LA chooses an operation such as a_i at nth stage and receives a favorable response from the environment, the probability of a_i operation being $p_i(n)$ increases and the probability of other operations is reduced. During this process, the automata learn to choose the optimal action. Cellular Learning Automata (CLA) is identified as a network of Learning Automata.

A common way to represent documents is to display text documents as a set of *n*-grams. Document words are extracted, and eventually a set of words is available that represents the entire document. This set defines the space in which each individual word is considered a dimension [45]. Each word in the document is assigned a weight that specifies the importance of that word for the document's separation. We use irregular uniform (IU) and uniform CLA (UCLA) algorithms as well as a set of union rules that are added in order to keep the order of words (the order in *n*-grams). The applied CLA has one (or more) LA in each cell, which chooses one action among its allowed actions at each time step. In this research, we used L_{R-I} linear learning model for all LAs. Each cell of CLA is assigned to the items that are available in each row of the dataset. At first, each cell is considered a neighborhood center, and union rules are applied to it. If the cell(s) are repeated more than the threshold number, they are sent to the output as a joint *n*-gram. The algorithm steps are as follows [26]:

1) Save each sentence of the document in a data record.
2) Create a matrix of *n*-grams such that the number of rows and columns is the number of sentences in the document. The diagonal elements are zero and other elements are calculated based on *n*-gram junctions.
3) Calculate *n*-gram relations between sentences with CLA (n> 1):
    a. Each word in dataset row is located in each cell as a rule of union.
    b. Each transaction of the neighborhood base union rules.
    c. If the current transaction repeats the previous transaction, the neighborhood is reinforced.
    d. Repeat each time will reward each cell with an *n*-gram output.
    e. If the reinforcement rate exceeds the threshold, they are sent to the output as a high frequency sequence.
4) Update: Neighbors and *n*-gram matrices are updated.
5) Validation: Steps will continue until the entire data set transaction is read and performed on the CLA and *n*-gram matrices are extracted.

2-5- Evaluation Method

Summaries created by the proposed method was evaluated by the ROUGE measurement toolkit [46]. ROUGE is an instrument which estimates the quality of the summary by tallying the covering units between the reference outline and the competitor summary. This takes place by n-gram review between a produced summary and a lot of reference summaries. Eq. (8) demonstrates the figuring of this measure.

$$\mathrm{ROUGE} = \frac{\sum_{S\in \mathrm{Summ_{ref}}} \sum_{n-\mathrm{gram}\in S} \mathrm{Count_{match}}(n-\mathrm{gram})}{\sum_{S\in \mathrm{Summ_{ref}}} \sum_{n-\mathrm{gram}\in S} \mathrm{Count}(n-\mathrm{gram})} \quad (8)$$

Table 1: Summary Implementation Methods

		Precision	Recall	F-measure
1	Simple Summarization	0.167654	0.241555	0.203880
2	Ontology	0.183167	0.269166	0.214633
3	Ontology + K-means	0.184629	0.320000	0.244738
4	Ontology + K-means + CLA	0.199990	0.325816	0.284305

Journal of Theoretical and Applied Information Technology
30th September 2023. Vol.101. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3- EXPERIMENTAL RESULTS

3-1- Data Set

The Document Understanding Conference (DUC) for 2002 was used to evaluate the proposed method [1]. The reason for the selection is to compare the proposed method with previous work. The DUC 2002 includes two sets: a) a training set and b) a test set. The training set consists of 30 sets, each containing approximately 10 documents, and each summary is written by 10 human experts. The test suite contains 30 sets of documents. For the body language, natural language projects used at Stanford University were used [2]. The ontology information was also extracted from OLIA ontologies [3]. For training and inferring test document vectors using paragraph vectors or doc2vec [4].

3-2- Results

ROUGE-2 was used to evaluate the experimental results[1]. The parameters obtained by the evaluation method are F-measure, Precision, Recall, and the obtained values of the parameters for comparison are presented in Table 1. In this way, the summary score was obtained with the least executable state, and then step-by-step was executed to allow for comparisons:

Table 2: Comparison of Text Summary Techniques

	Simple Summarization	Ontology	K-means	CLA
1	×	-	-	-
2	-	×	-	-
3	-	×	×	-
4	-	×	×	×

The results of the summarization scores in Table 2 and Fig. 4 are shown.

Also for comparison, the results were compared with those of Abbasi et al. [26] because they used the CLA method to summarization, It should be noted that the database used by these researchers was DUC 2002. Table 3 shows the results of the comparison of the present study with that of Abbasi et al., Which shows that in the case of Ontology + K-means + CLA with the results of Abbasi et al. Are 19.26%.

Table 3: Comparison of the results of the present study

	present study	Abbasi et al.	Percent
Simple Summarization	0.203110	0.2291	-12.79595258
Ontology	0.215715	0.2291	-6.204737215
Ontology + K-means	0.244913	0.2291	6.456406913
Ontology + K-means + CLA	0.283765	0.2291	19.26417889

4- CONCLUSION

In this paper, a new model based on ontology, unsupervised learning, and cellular learning automata for the text summarization problem is proposed. Using ontology, the meaning of each sentence can be examined based on the related concepts defined in ontology, and then each sentence can be assigned to a cluster related to these concepts. Given that labeled data is scarce and expensive in many problems, unsupervised learning is used as an effective and economical method for learning complex models. In the text summarization problem, unsupervised learning can be used to divide sentences into clusters with similar themes and concepts based on existing similarities between them. This can help with sentence clustering, and ultimately, using one of the representative sentences in each cluster, a summary of the text can be generated. By using cellular automata, the optimization of summarized sentences can be performed. In this model, considering the concepts and relationships between sentences, using optimization algorithms, summarized sentences can be produced in a way that important and key information from the text is preserved, resulting in the best possible summary.

In reviewing the results, the paper shows that the quality of the summarized text is improved by an average of 19.26% compared to other works done in the literature. However, this value can somehow be considered as an improvement, but it seems that further improvement in the quality of summaries should be made to be able to compete with existing methods in the field of text summarization. By the way, this article is only dedicated to English texts and cannot be used for other languages. Also, part of

[1] https://github.com/kavgan/ROUGE-2.0

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

this paper's model uses K-means unsupervised learning, which may not improve significantly for data with complex and skewed distributions. In general, this paper is considered as an innovation in the field of text summarization and can be useful for future research and improvement of existing methods in this field. However, to improve the quality of summaries, methods with higher accuracy and greater generalizability should be used.

5- FUTURE RESEARCH

In the context of learning for contextual data, the main focus is on deep recursive networks, supervised, unsupervised, and semi-supervised learning. In semi-supervised learning methods, unlabeled data is used together with labeled data to improve learning accuracy. To continue this work, supervised learning methods such as support vector machines and semi-supervised learning methods with CLA are suggested.

REFERENCES

[1] R. Ferreira *et al.*, "Assessing sentence scoring techniques for extractive text summarization," *Expert systems with applications,* vol. 40, no. 14, pp. 5755-5764, 2013.

[2] Z. Xiang, Z. Schwartz, J. H. Gerdes Jr, and M. Uysal, "What can big data and text analytics tell us about hotel guest experience and satisfaction?," *International Journal of Hospitality Management,* vol. 44, pp. 120-130, 2015.

[3] K. Ganesan, C. Zhai, and J. Han, "Opinosis: A graph based approach to abstractive summarization of highly redundant opinions," 2010.

[4] R. Brandow, K. Mitze, and L. F. Rau, "Automatic condensation of electronic publications by sentence selection," *Information Processing & Management,* vol. 31, no. 5, pp. 675-685, 1995.

[5] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of research and development,* vol. 2, no. 2, pp. 159-165, 1958.

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

[7] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," *Advances in Automatic Summarization,* pp. 55-60, 1999.

[8] Y.-I. Song, K.-S. Han, and H.-C. Rim, "A term weighting method based on lexical chain for automatic summarization," in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2004, pp. 636-639: Springer.

[9] Y. HaCohen-Kerner, Z. Gross, and A. Masa, "Automatic extraction and learning of keyphrases from scientific articles," in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2005, pp. 657-669: Springer.

[10] E. Villatoro-Tello, L. Villaseñor-Pineda, and M. Montes-y-Gómez, "Using word sequences for text summarization," in *International Conference on Text, Speech and Dialogue*, 2006, pp. 293-300: Springer.

[11] N. Shafiq, I. Hamid , M. Asif , Q. Nawaz , H. Aljuaid , H. Ali, "Abstractive text summarization of low-resourced languages using deep learning". PeerJ Comput. Sci. 9:e1176, 2023. http://doi.org/10.7717/peerj-cs.1176

[12] S. Song, H. Huang, and T. Ruan, "Abstractive text summarization using LSTM-CNN based deep learning," *Multimedia Tools and Applications,* vol. 78, no. 1, pp. 857-875, 2019.

[13] R. M. Alguliyev, R. M. Aliguliyev, N. R. Isazade, A. Abdi, and N. Idris, "COSUM: Text summarization based on clustering and optimization," *Expert Systems,* vol. 36, no. 1, p. e12340, 2019.

[14] A. Khan *et al.*, "Abstractive text summarization based on improved semantic graph approach," *International Journal of Parallel Programming,* vol. 46, no. 5, pp. 992-1016, 2018.

[15] M. J. Mohan, C. Sunitha, A. Ganesh, and A. Jaya, "A study on ontology based abstractive summarization," *Procedia Computer Science,* vol. 87, pp. 32-37, 2016.

[16] X. Zhang, G. Cheng, and Y. Qu, "Ontology summarization based on rdf sentence graph," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 707-716: ACM.

[17] E. Baralis, L. Cagliero, S. Jabeen, A. Fiori, and S. Shah, "Multi-document summarization based on the Yago ontology," *Expert Systems with Applications,* vol. 40, no. 17, pp. 6976-6984, 2013.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[18] P. Hípola, J. A. Senso, A. Leiva-Mederos, and S. Domínguez-Velasco, "Ontology-based text summarization. The case of Texminer," *Library Hi Tech,* vol. 32, no. 2, pp. 229-248, 2014.

[19] R. Barzilay and L. Lee, "Learning to paraphrase: an unsupervised approach using multiple-sequence alignment," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, 2003, pp. 16-23: Association for Computational Linguistics.

[20] K. Filippova, "Multi-sentence compression: Finding shortest paths in word graphs," in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010, pp. 322-330: Association for Computational Linguistics.

[21] Y. Mehdad, G. Carenini, F. Tompa, and R. Ng, "Abstractive meeting summarization with entailment and fusion," in *Proceedings of the 14th European Workshop on Natural Language Generation*, 2013, pp. 136-146.

[22] J. Niu, H. Chen, Q. Zhao, L. Su, and M. Atiquzzaman, "Multi-document abstractive summarization using chunk-graph and recurrent neural network," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1-6: IEEE.

[23] M. Khadhraoui, H. Bellaaj, M. B. Ammar, H. Hamam, M. Jmaiel," Survey of BERT-Base Models for Scientific Text Classification: COVID-19 Case Study". Appl. Sci. 2022, 12, 2891. https://doi.org/10.3390/app12062891

[24] A. Jain, A. Arora, J. Morato, D. Yadav and K. V. Kumar," Automatic Text Summarization for Hindi Using Real Coded Genetic Algorithm". Appl. Sci. 2022, 12, 6584

[25] A. Patil, K. Pharande, D. Nale, and R. Agrawal, "Automatic Text Summarization," *International Journal of Computer Applications,* vol. 109, pp. 18-19, 01/16 2015.

[26] R. Abbasi-ghalehtaki, H. Khotanlou, and M. Esmaeilpour, "Fuzzy evolutionary cellular learning automata model for text summarization," *Swarm and Evolutionary Computation,* vol. 30, pp. 11-26, 2016.

[27] P. Mitra, G. Wiederhold, and M. Kersten, "A graph-oriented model for articulation of ontology interdependencies," in *International Conference on Extending Database Technology*, 2000, pp. 86-100: Springer.

[28] J. N. Liu, Y.-L. He, E. H. Lim, and X.-Z. Wang, "A new method for knowledge and information management domain ontology graph model," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 43, no. 1, pp. 115-127, 2013.

[29] N. Nazari and M. A. Mahdavi, "A Survey on Automatic Text Summarization", Journal od AI and Data Mining, Vol. 7,No. 1, pp. 121-135, 2019

[30] A. P. Widyassari, S. Rustad, G. F. Shidik, E. Noersasongko, A. Syukur, A. Affandy, D. I. Setiadi, "Review of Automatic Text Summarization Techniques & methods", Journal of king Soud University- Computer and Information Sciences, Vol. 34, No. 4, pp. 1029-1046, 2022.

[31] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management,* vol. 24, no. 5, pp. 513-523, 1988.

[32] M. R. J. P.-S. Bautista-Zambrana and B. Sciences, "Creating corpus-based ontologies: a proposal for preparatory work," vol. 212, pp. 159-165, 2015.

[33] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition,* vol. 36, no. 2, pp. 451-461, 2003.

[34] A. Fahim, A. Salem, F. A. Torkey, and M. Ramadan, "An efficient enhanced k-means clustering algorithm," *Journal of Zhejiang University-Science A,* vol. 7, no. 10, pp. 1626-1633, 2006.

[35] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849-856.

[36] J. Kumar, R. T. Mills, F. M. Hoffman, and W. W. Hargrove, "Parallel k-means clustering for quantitative ecoregion delineation using large data sets," *Procedia Computer Science,* vol. 4, pp. 1602-1611, 2011.

[37] K. E. Dierckens, A. B. Harrison, C. K. Leung, and A. V. Pind, "A Data Science and Engineering Solution for Fast K-Means

Clustering of Big Data," in *Trustcom/BigDataSE/ICESS, 2017 IEEE*, 2017, pp. 925-932: IEEE.

[38] A. Algergawy, S. Massmann, and E. Rahm, "A clustering-based approach for large-scale ontology matching," in *East European Conference on Advances in Databases and Information Systems*, 2011, pp. 415-428: Springer.

[39] I. Jordanov and R. Jain, *Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2010.

[40] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition,* vol. 5, no. 2, pp. 199-220, 1993.

[41] N. Guarino, "Formal ontology, conceptual analysis and knowledge representation," *International journal of human-computer studies,* vol. 43, no. 5-6, pp. 625-640, 1995.

[42] M. Yoshioka, Y. Umeda, H. Takeda, Y. Shimomura, Y. Nomaguchi, and T. Tomiyama, "Physical concept ontology for the knowledge intensive engineering framework," *Advanced engineering informatics,* vol. 18, no. 2, pp. 95-113, 2004.

[43] L. Hennig, W. Umbrath, and R. Wetzker, "An ontology-based approach to text summarization," in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2008, vol. 3, pp. 291-294: IEEE.

[44] J. Von Neumann and A. W. Burks, "Theory of self-reproducing automata," *IEEE Transactions on Neural Networks,* vol. 5, no. 1, pp. 3-14, 1966.

[45] G. Salton, E. A. Fox, and H. Wu, "Extended Boolean information retrieval," Cornell University1982.

[46] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Text Summarization Branches Out,* 2004.

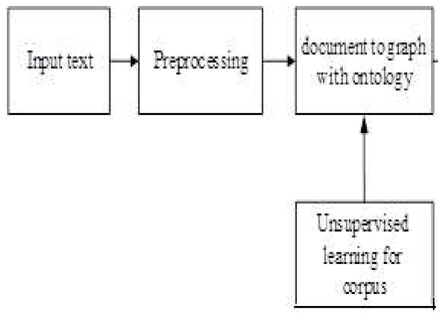

Figure 1: The Proposed Method For Text Summarization

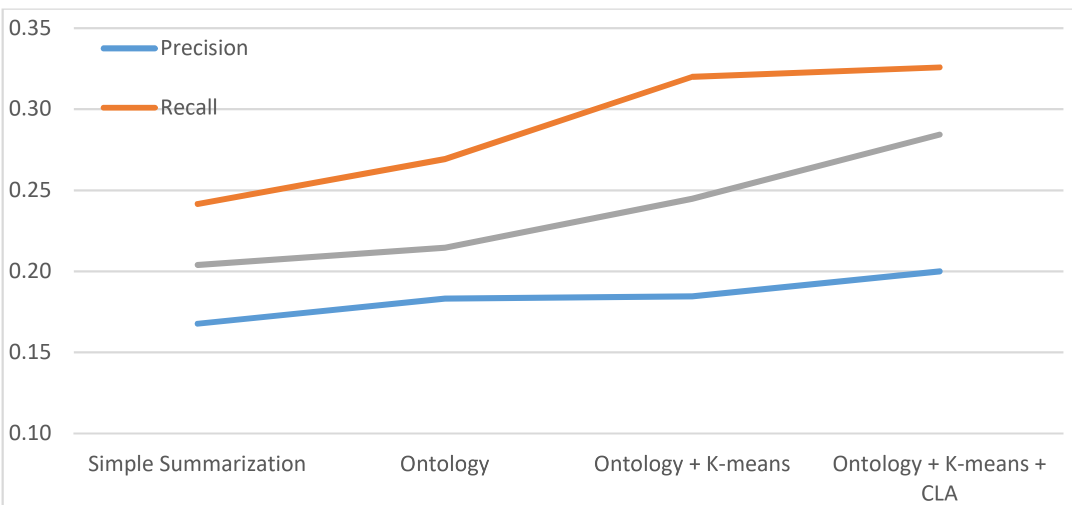

Figure 4: Comparison Of Text Summary Techniques