Journal of Information Systems and Telecommunication
Vol.10, No.1, January-March 2022, 68-79

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

A Survey on Multi-document Summarization and Domain-Oriented Approaches

Mahsa Afsharizadeh[1], Hossein Ebrahimpour-Komleh[1*], Ayoub Bagheri[2], Grzegorz Chrupala[3]

[1]. Department of Computer and Electrical Engineering, University of Kashan, Iran
[2]. Department of Social Sciences, Utrecht University, Utrecht, The Netherlands
[3]. Department of Cognitive Science and Artificial Intelligence, Tilburg University, The Netherlands

Received: 01 Jun 2021 / Revised: 06 Sep 2021/ Accepted: 06 Oct 2021
DOI: https://doi.org/10.52547/jist.16245.10.37.68

Abstract

Before the advent of the World Wide Web, lack of information was a problem. But with the advent of the web today, we are faced with an explosive amount of information in every area of search. This extra information is troublesome and prevents a quick and correct decision. This is the problem of information overload. Multi-document summarization is an important solution for this problem by producing a brief summary containing the most important information from a set of documents in a short time. This summary should preserve the main concepts of the documents. When the input documents are related to a specific domain, for example, medicine or law, summarization faces more challenges. Domain-oriented summarization methods use special characteristics related to that domain to generate summaries. This paper introduces the purpose of multi-document summarization systems and discusses domain-oriented approaches. Various methods have been proposed by researchers for multi-document summarization. This survey reviews the categorizations that authors have made on multi-document summarization methods. We also categorize the multi-document summarization methods into six categories: machine learning, clustering, graph, Latent Dirichlet Allocation (LDA), optimization, and deep learning. We review the different methods presented in each of these groups. We also compare the advantages and disadvantages of these groups. We have discussed the standard datasets used in this field, evaluation measures, challenges and recommendations.

Keywords: Multi-document Summarization; Single Document Summarization; Extractive; Abstractive; Domain-Oriented; ROUGE.

1- Introduction

Unlike in the past, there was often not enough data on every issue, but today we are faced with the issue of information overload. Obtaining the most important information from a huge amount of data is a time-consuming and difficult task. Various fields such as natural language processing (NLP), text mining and artificial intelligence were used to provide a solution to this problem. Automatic text document summarization is an important solution to this problem. Text summarization obtains a short and compact gist that preserves the main concepts of the original text. So the user can understand the concept of a long text in the form of a brief summary.

According to different perspectives, text summarization methods can be divided into several categories.

From the summary producing perspective, summarization is divided into two categories: extractive and abstractive. Extractive summarizing merges the extracts from the original text and presents them as summaries. While abstractive summarizing paraphrases the text and generates new sentences [1]. Summarization can be generic or query-based. The generic summary is about the whole text while the query-based summary is about the query being asked [2]. In terms of the number of texts, the summarization is divided into two categories: single document summarization (SDS) and multi-document summarization (MDS). The purpose of summarizing a single document is to produce a summary of one text, while the purpose of multi-document summarizing is to produce a short, relevant summary of a set of several textual documents related to a similar topic [3]. In terms of the domain of the input text, summarization techniques are divided into two categories: domain-oriented and domain-independent techniques. Domain-oriented methods perform summarization for texts related to a specific domain, for example, medicine, law, etc. While domain-

✉ Corresponding Author
ebrahimpour@kashanu.ac.ir

independent summarization techniques produce summaries without considering the domain of the input text.

For many years, major research was performed on SDS. Research in this field is still ongoing. MDS is necessary to apply summarization at larger scales. The following example illustrates the need for MDS. A user searches for a specific topic on the World Wide Web. This search will lead to the retrieval a lot of related documents. There is probably a lot of similar information between these related documents. In this situation, the performance of a single document summarizer on each of these documents leads to the production of multiple summaries with plenty of redundant information [4]. Therefore, a single document summarizer cannot fulfill the main goal of the summarization task, which is generating a summary with minimum redundancy and maximum relevancy [5]. Multi-document summarization has emerged as an effective solution to such a situation.

Research in the field of automatic text summarization began with SDS and moved to MDS after a while. In recent years, various approaches to MDS have been proposed.

Sometimes text documents are related to a specific domain, such as medicine, law, terrorism, etc. Applying the usual text summarization methods to such documents will not produce satisfactory results. Because text documents that are written in a particular domain often have a certain structure and characteristics. Domain-oriented summarizers encounter more challenges. Such systems, in addition to commonly used summarization techniques, use the structure and specific characteristics of that domain to identify deeper information from text documents, which results in generating more efficient summaries.

This paper is organized as follows: section 2 presents definitions, applications, and categorizations of multi-document summarizers by different authors. Section 3 provides a new categorization for MDS methods in six categories: machine learning, clustering, graph, LDA, optimization, and deep learning-based approaches. Section 4 describes domain-oriented summarization. Section 5 presents the datasets and standard measures for evaluating a summarization system. Section 6 explains challenges and recommendations in the field of MDS. Finally, section 7 concludes the paper.

2- Multi-document Summarizer

With the increase in the amount of information on web pages, finding the desired information has become a difficult issue. For a given topic, there may be hundreds of documents that are not necessarily related to it. In order to find the related information, a user needs to search among all the documents. This results in a huge amount of information, with a lot of time and effort. To cope with this problem, automatic text summarization plays a vital role [6]. Automatic summarization of text documents is a method for producing a compressed version of the original document.

So far different categorizations have been presented in the context of text summarization. Yosefi-Azar and Hamey categorized text summarization techniques into three types [7]:

- Classic approaches
- Machine learning-based approaches
- Artificial neural network-based approaches.

The initial summarization methods in the classical approaches were based on the frequency of words occurring in the text. The sum of the frequencies of the words that make up a sentence can be considered as a score to indicate the importance of that sentence in the whole text. In this set of methods, other word level and sentence level features like key phrases (for example title and heading words) and the position of sentences were also used. Cluster-based and graph-based methods can also be considered as classical methods.

Machine learning-based summarizers are trainable systems which learn how to tune their parameters to extract salient content. The set of features in these techniques is often the same as the classical methods. Machine learning summarizers include several methods such as Decision Tree, Hidden Markov Model (HMM), Support Vector Machine (SVM), and Support Vector Regression (SVR).

Artificial Neural Networks (ANN) are the other trainable systems that simulate the structure of the human brain. A neural network can be trained to learn the features of the informative sentences. Different types of shallow and deep neural networks are used to summarize the text. Deep neural networks have promising results in summarizing the text.

Maria Fuentes Fort describes the methods used in automatic summarizing from two perspectives, classical perspective and multi-task perspective [8]. Summarization categorization according to Fuentes Fort's view is shown in Fig 1.

In the classical perspective, automatic summarization can be divided into three levels based on the level that the summarizer processes texts: surface level, entity level, and discourse level.

- **Surface Level:** Surface level techniques use shallow linguistic features to display information. The combination of these features produces a salience function for distinguishing important text information. Shallow linguistic features used in surface-based methods can be divided into four groups: term frequency-based features, location-based features, bias-based features, and cue words features.
- **Entity-Level:** Entity-level summarization methods provide an internal representation from the text to determine the salience by modeling the text entities (for example simple words, compound words, named entities, and terms) and relationships (such as similarity, proximity, co-occurrence, co-reference, etc.).
- **Discourse Level:** Discourse level methods model the global structure of the text. Cohesion and coherence are two main features in text discourse structure.

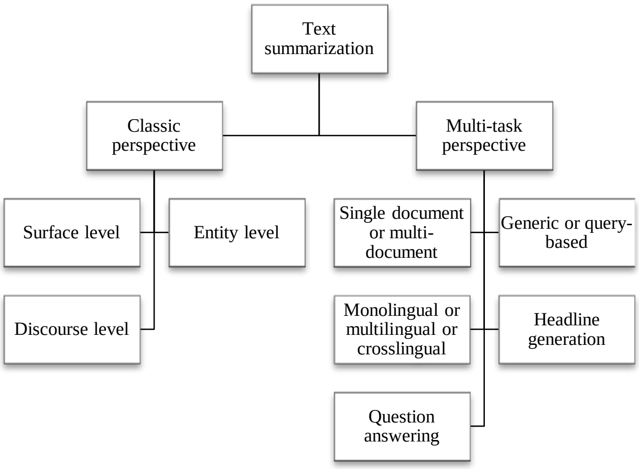

Fig. 1 Summarization categorization according to Fuentes Fort's view [8]

- Multi-task perspective divides methods by different summarization tasks.
- **SDS versus MDS:** Based on the number of documents, summarization methods are divided into SDS or MDS methods.
- **Query-Based versus Generic Summarization:** If a user-specific information type is required, a query-based summarization is provided. While in generic summarization, all relevant topics should be included in the summary.
- **Monolingual/Multilingual versus Cross-lingual Summarization:** Based on language coverage, systems are divided into three categories: monolingual, multilingual, and cross-lingual. In monolingual and multilingual summarizing systems, the language of the input documents and the summary are the same. The first one deals with only one language, but the latter works in several languages. In contrast, cross-lingual systems can process input documents in different languages, as well as summarize them in different languages.
- **Headline Generation:** In the task of headline generation, the purpose is generating a very short summary as a headline for a text.
- **Question Answering:** Question answering systems deal with a question and a document (or a set of documents) which seems to be relevant to the question. The task is to create a short summary of the document answering that question.

2-1- Definition of Multi-document Summarization

MDS is a way to display the main content of a set of documents with a similar topic by a short text by including important and relevant information and filtering out the redundant information. Two prominent approaches in summarizing multiple documents are extractive and abstractive summarization. Extractive systems aim to extract prominent sentences from documents, while abstractive summarization systems aim to paraphrase the contents of documents to generate a new shortened text [9].

2-2- Applications of Multi-document Summarizers

Text summarization has many applications in today's world. For example, the production of a summary of various E-books or scientific articles, the production of summaries of patients' medical information, generation a site summary (what search engines like Google do), summarizing product reviews, student responses to classroom questionnaires, and a series of news articles about a specific topic [10].

Of course, these are only a few examples of the many uses of this topic in today's society.

2-3- Categorizations on MDS

Some researchers have provided categorizations for MDS summarizing methods. Joshi and Kadam have divided MDS into three categories: cluster-based approaches, ranking-based approaches, and LDA-based approaches [5].

Tabassum and Oliveria have divided MDS into five types from another perspective: feature-based approaches, domain-specific (ontology-based) approaches, cognitive-based approaches, event-based approaches, and discourse-based approaches [11].

Shah and Jivani have done another categorization on MDS methods [12]. From their point of view, MDS methods can be classified into four categories: graph-based approaches, cluster-based approaches, term frequency-based methods, and Latent Semantic Analysis (LSA).

In another study, Tandel et al. classified MDS methods into three categories: cluster-based approaches, topic-based approaches, and lexical chain approaches [13].

Gupta and Lehal categorized summarization methods into these categories:

Various categorizations on MDS presented by different researchers [5], [11], [12], and [13] are shown in Fig 2.

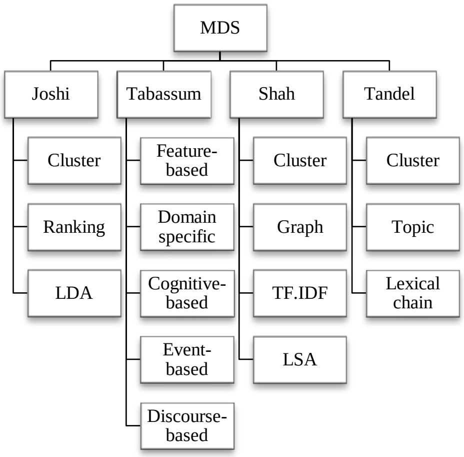

Fig. 2 Various categorizations on MDS Source: [5], [11], [12], and [13]

3- Proposed Categorization on MDS

We have also proposed a more comprehensive categorization for MDS methods by reviewing the previous works in this field. The Categorization presented in this paper for MDS methods is shown in Fig 3. We categorize MDS methods into six categories: machine learning, clustering, graph, LDA, optimization, and deep learning.

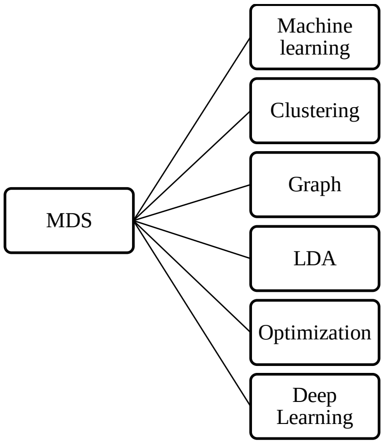

Fig. 3 Categorization of MDS approaches

3-1- Machine Learning-based Approaches:

Machine learning methods are widely used in the summarization process. For example SVM [14], ANN [15], and Decision Tree [16] are various types of machine learning methods used in summarizing.

One of the SVM-based MDS techniques was proposed by Chali et al [14]. They used an ensemble of SVMs. Each SVM made its own prediction for an unseen sentence. The SVM ensemble then combined these predictions based on the weighted averaging technique and produced the final prediction. Four different SVMs were used to rank the sentences, and finally the top-ranked sentences were selected for the final summary.

Neural networks have also been used to summarize multiple documents. One such study was conducted in 2016 by Ma et al. [15]. They proposed an unsupervised multi-document summarization framework based on a neural document model. In this method, a neural document model in the multi-document summarization task and a document level reconstruction framework called DocRebuild are proposed. Neural document model tries to display the semantic content of documents using low-dimensional vector representations. In this method, two types of unsupervised neural document models called Bag-of-Words (BOW) and Paragraph Vector (PV) are used to represent the semantic content of documents. DocRebuild reconstructs documents with summary sentences through a neural document model and selects summary sentences to minimize reconstruction errors.

The decision tree has also been used to summarize multiple documents. Ou et al. used decision trees to design their multi-document summarization system [17]. They proposed a multi-document summarization system for sociology dissertation abstracts. Their proposed system performs a discourse parsing after receiving and pre-processing a set of related dissertation abstracts on a specific topic. In this step, using a decision tree classifier, each sentence is classified into one of five predefined sections: *background*, *research objectives*, *research methods*, *research results* and *concluding remarks*. Then in the information extraction stage, important concepts are extracted from each of the sections using pattern matching. Then, in the information integration stage, the obtained information is clustered so that similar concepts are in the same clusters. Finally, based on this integrated information, a final summary is generated.

3-2- Clustering-based Approaches

Clustering-based methods have also been used to summarize the text. These methods are able to identify various topics raised in the texts. One of the works in this field was done by Gupta and Siddiqui [18]. Their method was multi-document and query-based.

This method first generated an SDS for each text. All SDSs were then combined and a clustering was applied to the set of all sentences. In the clustering process, syntactic and semantic similarities between sentences were considered. From each cluster, the most important sentence was selected, and the set of selected sentences was sorted based on their location in the main document and constitute the final summary.

Cai and Li proposed an integrated approach using ranking through clustering for MDS [19]. Applying clustering to documents leads to the production of a number of topic themes. Each theme is displayed by a cluster of highly related sentences. For each topic theme, the term ranks in that topic theme should be different and distinct from term ranks in other topic themes. Most existing cluster-based summarization systems apply clustering and ranking methods individually, which results in incomplete or sometimes biased results. In Cai and Li’s approach clustering results are used to improve or refine the sentence ranking results. The main idea of this method is that the ranking distribution of sentences in each cluster should be different. In their approach ranking and clustering simultaneously update each other, and the performance of both is improved.

3-3- Graph-based Approaches

Graph-based methods for MDS are widely used to extract top sentences for summaries. Al-Dhelaan presents a simple star graph for MDS called StarSum [20]. StarSum is a star bipartite graph that models sentences and their topic signature phrases. This method extracts sentences that guarantee the diversity and coverage, both of which are essential for MDS. The diversity is guaranteed by splitting the StarSum graph into different components and using top sentences from each of the different components. Coverage is guaranteed by ranking sentences by their degree of connection to other topic sentences and phrases.

Khan et al. present a clustered semantic graph approach for multi-document abstractive summarization [21]. Most of the existing graph-based methods rely on the bag of words method, which treats sentences as a bag of words and relies on a content similarity measure. The main limitation of the bag of words method is that it does not consider the relationship between words. This paper proposes a clustered semantic graph-based method for abstractive MDS. This method uses Semantic Role Labeling (SRL) to extract the semantic structures called Predicate Argument Structure (PAS). Pairwise PASs are compared based on the linear semantic similarity measure to create a semantic similarity matrix. This matrix is represented as a semantic graph. PASs are vertices of the graph. The edges represent the semantic similarity weight between the vertices. For content selection, graph nodes (PASs) are ranked based on the modified graph-based ranking algorithm. Maximal Marginal Relevance (MMR) is applied to redundancy reduction. PAS representatives with the highest salience score are selected from each cluster and are fed to language generation to generate summary sentences.

Glavas and Snajder proposed event graphs for MDS [22]. They first created an event-based document representation of the text. This representation is an event graph that contains information about the events described in the text. Event mentions are then extracted from the sentences. Then event graph information is used to understand the importance of the events in the sentence and the relationships between the events. Each sentence gets a score according to the importance of the events presented in it and is selected according to these scores to attend the summary.

3-4- LDA-based Approaches

LDA dates back to 2003 [23]. It is a generative probabilistic model. LDA in natural language processing can do topic discovery. The basic idea of the LDA is that documents are displayed as random mixtures on latent topics, where each topic is a distribution of words. A set of documents has a probabilistic distribution of topics so that a particular document probably contains some topics more than others. Terms within a topic also have their own probability distribution. That is, in a topic, some specific terms are used much more than others. In LDA, both sets of probabilities in the training phase are calculated using the Bayesian methods and the expectation maximization algorithm. Some researchers have used LDA to perform MDS. LDA-based extractive summarization methods, after discovering the hidden topics in the text, select the most important sentences in each topic as a representative of that topic to participate in the summary.

Roul [24] proposed a topic-based model for extractive multi-document text summarization. The model identifies the number of independent topics using LDA and three probabilistic models. The probabilistic methods Hellinger's distance, Jensen Shannon Divergence, and KL divergence are used to compute the similarity between each of the topic pairs. Then LDA technique is used again to reduce a large set of n sentences to a smaller set while maintaining the important information. The representative sentence from each topic is selected and arranged by the corresponding topic importance to appear in the summary.

One of the other researches in this field has been done by Na et al [25]. They proposed a method called Titled-LDA. Titled-LDA is an extended version of LDA that does both title topic modeling and content topic modeling. The two models are then combined to create a new mixture topic model. In the mixing step, a weight is assigned to each of the two distributions. These weights are learned by an adaptive asymmetric learning algorithm. Titled-LDA consists of five steps: (1) Extracting title and content from each document. (2) Obtaining title and content topic distributions. (3) Combining topic models using adaptive asymmetric learning algorithm. (4) Calculating sentence scores based on mixture topic model. (5) Generate summaries based on sentence scores.

3-5- Optimization-based Approaches

The MDS task faces challenges such as redundancy, complementarity, and contradiction [26]. To generate informative extractive MDS, the most important set of sentences should be selected, avoiding redundancy and contradiction, and maintaining complementarity between them. Each of the phenomena can be considered as an objective function and optimization methods can be used to solve it.

Su et al. [27] proposed an optimization-based MDS called PoBRL (Policy Blending with maximum marginal relevance and reinforcement Learning). PoBRL considers three objectives: importance, non-redundancy and length for the summarization task. This multi-objective optimization task is broken down into different sub-tasks, each of which is solved separately by reinforcement learning. The learned policies are then combined by PoBRL to produce the final summary.

Alguliev et al. consider MDS as an evolutionary optimization problem [28]. The sentence-to-document collection, summary-to-document collection and sentence-to-sentence relationships are used to select salient sentences from the document collection and reduce redundancy. They consider this problem as a discrete optimization problem. To solve the discrete optimization, a self-adaptive Differential Evolution (DE) algorithm has been created.

Sanchez-Gomez et al. provided an extractive MDS using a multi-objective artificial bee colony optimization approach [29]. The extractive MDS methods aim to obtain the main content of a data collection and simultaneously reduce redundant information. The paper analyzes this issue from the perspective of optimization. For this purpose, a Multi-Objective Artificial Bee Colony (MOABC) algorithm is proposed. The MDS problem needs to be optimized for more than one objective function and so Multi-Objective Optimization can be used for this purpose. MOABC has three types of bees that allow different search mechanisms for any bee. Employed bees maintain current solutions. Onlooker bees allow exploitation of the best solutions ever found. Scout bees eliminate stagnated solutions and allow exploration of the partially good solutions. This combination of exploration and exploitation mechanisms provides an effective way for MDS.

John et al. formulated extractive MDS as population-based multi-criteria optimization [30]. They consider three objective functions for determining an optimal summary: maximum relevance, diversity, and novelty. For this purpose, both syntactic and semantic aspects of the document are considered. The semantic aspect is considered through LSA techniques and Negative Matrix Factorization. In each iteration of the algorithm, three candidate summaries are identified that maximize the value of the objective functions and create the final optimal summary. Table 1 shows comparison matrix of machine learning, clustering, graph, LDA, and optimization approaches.

3-6- Deep Learning-based Approaches

In recent years, deep learning has gained significant results in many areas of NLP, including text summarization. Many scholars have focused their attention on deep learning methods for MDS.

Afsharizadeh et al. [31] proposed an extractive summarization using Recurrent Neural Networks (RNN) and coreference resolution procedure. The model stores coreference information in the form of coreference vectors. A three-layer Bidirectional Long Short Term Memory (Bi-LSTM) computes sentence representations using the embedding vectors of their constituent words. The sentence representations are then enriched using the coreference vectors.

Zhang et al. provide a multiview convolutional neural network for extractive MDS [32]. In this paper, an extended CNN was used to obtain sentence features and to rank sentences. Multiview learning was added to the model to improve CNN's ability to learn. Three CNN networks are used to generate a summary. Each CNN has a salience score for each sentence. Then these scores are combined to get the final score for the sentences.

Cao et al. develop a ranking framework for the recursive neural network to rank sentences in the MDS [33]. This article formulated the sentence ranking task as a hierarchical regression process. Recursive neural networks are used to learn auto-ranking features: learned features supplement hand-crafted features to rank sentences. Finally, ranking score sentences are used to effectively select informative and non-redundant sentences. Zhong et al. present a query-oriented unsupervised MDS through a deep learning model [34].

Table 1: Comparison matrix of machine learning, clustering, graph, LDA, and optimization approaches

Work	Category	Purpose	Dataset	Results
A SVM-based ensemble approach to multi-document summarization [14]	Machine learning: SVM	Using an ensemble of SVMs for MDS.	DUC 2007	**R-1:** 0.388 **R-2:** - **R-L:** 0.319 **R-SU:** 0.146
Unsupervised multi-document summarization framework based on neural document model [15]	Machine learning: NN	Using a document level reconstruction framework using neural document model for MDS.	DUC 2006 DUC 2007	**R-1:** 0.421, 0.434 **R-2:** 0.093, 0.105 **R-L:** - **R-SU:** 0.151, 0.162
Summarizing Indonesian text automatically using sentence scoring and decision tree [17]	Machine learning: Decision tree	Using a decision tree to classify each sentence into one of five predefined categories. Then extracting main concepts using pattern matching.	50 text documents	**R-1:** 0.580 **R-2:** - **R-L:** - **R-SU:** -
Multi-document summarization using sentence clustering [18]	Clustering	Applying clustering to a set of SDSs to make a multi-document summary.	DUC2002	**R-1:** 0.338 **R-2:** - **R-L:** - **R-SU:** -
Ranking through clustering: An integrated approach to multi-document summarization [19]	Clustering	Using ranking through clustering for MDS	DUC2004 DUC2005 DUC2006 DUC2007	**R-1:** 0.374, 0.364, 0.405, 0.416 **R-2:** 0.089, 0.073, 0.093, 0.120 **R-L:** - **R-SU:** -
StarSum: A Simple Star Graph for Multi-document Summarization [20]	Graph	Using a star bipartite graph that models sentences and their topic phrases to summarize multiple documents.	DUC2001	**R-1:** 0.523 **R-2:** 0.391 **R-L:** 0.511 **R-SU:** -
A clustered semantic graph	Graph	Using SRL to extract the semantic structures of the text	DUC2002	**R-1:** 0.400

Work	Category	Purpose	Dataset	Results
approach for multi-document abstractive summarization [21]		and represents them as a semantic graph. Nodes with the highest salience score are selected from each cluster generate summary sentences.		**R-2:** 0.099 **R-L:** - **R-SU:** -
Event graphs for information retrieval and multi-document summarization [22]	Graph	Proposing event graphs that contain information about the events described in the text for MDS.	DUC2002 DUC2004	**R-1:** 0.415, 0.405 **R-2:** 0.116, 0.107 **R-L:** - **R-SU:** -
Topic modeling combined with classification technique for extractive multi-document text summarization [24]	LDA	Using LDA and probabilistic models for text topic identification. Sentences with highest topic importance scores are selected for the summary.	DUC2002 DUC2006	**R-1:** 0.497, 0.429 **R-2:** 0.258, 0.094 **R-L:** - **R-SU:** -
Mixture of topic model for multi-document summarization [25]	LDA	Mixing title topic modelling and content topic modelling to create a new mixture topic model and using it for MDS.	DUC2002	**R-1:** 0.463 **R-2:** 0.182 **R-L:** 0.422 **R-SU:** 0.226
PoBRL: Optimizing Multi-document Summarization by Blending Reinforcement Learning Policies [27]	Optimization	Using a multi-objective optimization-based approach. Each objective is solved separately by reinforcement learning. The learned policies are then combined to produce the final summary.	MultiNews DUC2004	**R-1:** 0.465, 0.386 **R-2:** 0.173, 0.102 **R-L:** 0.424, 0.131 **R-SU:** -
Multiple documents summarization based on evolutionary optimization algorithm [28]	Optimization	Considering MDS as a discrete optimization problem. A self-adaptive DE algorithm is used to solve it.	DUC2002 DUC2004	**R-1:** 0.499, 0.393 **R-2:** 0.258, 0.112 **R-L:** 0.489, 0.396 **R-SU:** 0.287, 0.135
Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach [29]	Optimization	Proposing a multi-objective artificial bee colony optimization approach. The model has three types of bees. The combination of them provides an effective way for MDS.	DUC2002	**R-1:** - **R-2:** 0.312 **R-L:** 0.540 **R-SU:** -
Extractive multi-document summarization using population-based multicriteria optimization [30]	Optimization	Formulating extractive MDS as population-based multi-criteria optimization. Three objectives are used to consider both syntactic and semantic aspects of the text.	DUC2002 DUC2004 DUC2006	**R-1:** 0.548, 0.521, 0.325 **R-2:** 0.271, 0.171, 0.069 **R-L:** - **R-SU:** -

The proposed framework includes three parts: concept extraction, summary generation, and reconstruction validation. The deep Auto Encoder (AE) network is used for this purpose. The concept extraction part is the phase of encoding of the network and obtaining a compact representation of the concept. The reconstruction validation phase relates to the network decoding phase and attempts to reconstruct the inputs of the network. The summary generation step using dynamic programming generates a final summary of the candidate sentences.

Lakshmi and Rani provide a method for implementing MDS using deep learning and fuzzy logic [35]. Restricted Boltzmann Machines (RBM) are used to produce a shortened version of the document without losing its important information. First, the text is converted to a feature matrix, in which the rows are related to sentences and columns to features. The fuzzy classifier then assigns labels to sentences. A new feature matrix is formed by adding a column of labels to sentences. This feature matrix is considered as the input for the RBM which receives input from each row of this feature matrix. The RBM learns network weights by trying to reconstruct inputs. After generating a score for sentences, high ranked sentences are selected to be in the summary.

Table 2 shows a brief summary of deep learning-based approaches. Also, advantages and disadvantages of the methods used in MDS are shown in Table 3.

4- Domain-oriented Summarization

Sometimes a text is related to a specific domain. In this case, it often has a specific structure or characteristics unique to its domain type. Such characteristics help summarization algorithms to more accurately identify the most important information and provide a more detailed summary. For example, journal articles often have an abstract and conclusion section that contains the most important information about the text [36]. Multi-document summarizers have been applied in a wide range of domains, such as summarizing scientific articles, literary texts, blog posts, and patient data [37]. Accordingly, a multi-document summarizer can be a domain-oriented or domain-independent approach.

Some MDS systems are specifically designed for a particular genre of documents, for example, news articles about terrorism [38]. SUMMONS (SUMMarizing Online NewS Articles) was proposed as a summarizer of news articles [39]. The input of this system is a collection of generated templates by the MUC (Message Understanding Conference), which works on the domain of terrorism. Each template shows the information extracted from one or more articles. The templates are then compared and merged using different planning operators. Each operator combines a pair of templates for a new template. There are seven operators in SUMMONS including agreement, addition, contradiction, etc

Table 2: Comparison of some deep learning-based approaches

Paper Title	Purpose	Dataset	Results
Automatic Text Summarization of COVID-19 Research Articles Using Recurrent Neural Networks and Coreference Resolution [31]	Using a combination of RNNs and coreference resolution procedure for summarization. The model stores coreference information in the form of coreference vectors.	CORD19	**R-1:** 0.343 **R-2:** 0.116 **R-L:** 0.188 **R-SU:** 0.152
Multiview convolutional neural networks for multi-document extractive summarization [32]	Provide a multiview convolutional neural network for extractive MDS. Multiview learning was added to the model to improve CNN's ability to learn.	DUC2001 DUC2002 DUC2004 DUC2006 DUC2007	**R-1:** 0.359, 0.367, 0.390, 0.386, 0.409 **R-2:** 0.079, 0.090, 0.100, 0.079, 0.091 **R-L:** - **R-SU:** 0.131, 0.149, 0.136, 0.140, 0.153
Ranking with Recursive Neural Networks and Its Application to Multi-document Summarization [33]	Developing a ranking framework for the recursive neural network to rank sentences in the MDS.	DUC2001 DUC2002 DUC2004	**R-1:** 0.369, 0.379, 0.387 **R-2:** 0.078, 0.088, 0.098 **R-L:** - **R-SU:** -
Query-oriented unsupervised multi-document summarization via deep learning model [34]	Presenting a query-oriented unsupervised MDS through a deep learning model. The deep AE network is used for this purpose.	DUC2005 DUC2006 DUC2007	**R-1:** 0.375, 0.401, 0.429 **R-2:** 0.077, 0.092, 0.116 **R-L:** - **R-SU:** 0.134, 0.147, 0.168
Multi-document Text Summarization Using Deep Learning Algorithm with Fuzzy Logic [35]	Implementing MDS using RBM and fuzzy logic.	DUC2002	**R-1:** 0.550 **R-2:** - **R-L:** - **R-SU:** -

Table 3: Advantages and disadvantages of different approaches in the field of MDS

Method	Advantage	Disadvantage
Machine Learning	Using different Machine Learning methods such as Bayes Classifiers, Artificial Neural Networks, and SVM	**The main drawback of this method is obtaining a labeled dataset. Labeling sentences in documents is a time-consuming operation.**
Clustering	Clustering-based methods are suitable for texts with multiple different topics.	**K-means is the most famous clustering algorithm. One of its disadvantages is determining the appropriate value for k, i.e. the number of clusters. This method is also suitable for clusters with spherical shapes but not for non-convex shapes.**
Graph	These methods create a good visual representation of the text, which at a glance can identify the number of distinct topics in the text and the most important sentence in each topic.	**Choosing a way to score vertices is challenging.**
LDA	Ability to discover hidden topics in the text.	**The number of topics is fixed and must be determined in advance.**
Optimization	Sentences contained in the multi-document summary should be relevant, non-redundant, and non-contradict. They can be considered as objective functions. Then MDS is solved as an optimization problem. The goal is to find the best settings for summarization.	**This method may be slow since finding the optimal weights for objective functions is achieved after several iterations.**
Deep Learning	Ability to auto-learn features from raw data, suitable for large datasets	**These methods require a large amount of training data to learn model parameters.** **Possibility of vanishing and exploding gradient problems that cause the model not to be trained and therefore incorrect adjustment of parameters**

SUMMONS architecture consists of two main components: content planner and linguistic generator. Content planner generates a conceptual representation of the meaning of the text, and usually does not contain any linguistic information. The content planner determines what information in input templates should be included in the summary, and does this with a set of planning operators. An operator is used to link information in two different templates. A summary can be the result of applying a single operator. More complex summaries can be generated by multiple operators. The linguistic component consists of lexical chooser and sentence generator components.

The lexical chooser defines a high level sentence structure for each sentence. The sentence generator, using a large English grammar, meets the syntactic constraints, creates a syntactic tree, and linearizes the tree as a sentence. Radev suggested a Cross-Document Structure Theory (CST) that was a taxonomy of the relationship between the documents [40].

The CST concept is similar to the discourse structure in a single document. These cross-document relationships can be used in MDS, and some of them are a direct descendant of the ones used in SUMMONS.

When the input of the summarization system is related to a particular domain, conventional summarization methods may not be appropriate. When input documents include specialized information in a particular domain, they often have a specific structure and characteristics. These characteristics can help summarization algorithms to more accurately identify important information [36]. For example, journal articles often have a conclusion section that includes the key information of the article and contains important information for the summary. Certain domains like medical or law may have specific requirements for the type of information that is required in a summary. Such domains may also have resources that can help the summarization process. In the following, a number of summarization methods are reviewed in the medical domain.

Conventional summarization methods are not easily applicable to certain domains, such as the medical domain. In this domain, summarizing algorithms, with the precise use of specific medical definitions, have valuable applications such as helping clinicians in their treatment cycle, reviewing the latest research on a particular patient, or helping patients and their families with information about the disease. Medical articles have a specific structure that algorithms take. In addition, there are also extensive knowledge resources available in the medical field. The final users of medical summarizing systems are healthcare providers and consumers, who both can access information of interest through the Internet. In the medical community, the number of journals related to even a single field is very high and it is difficult for physicians to know about all the new results reported in their specialized fields. Similarly, patients and their family members who need information about their particular illness face a huge amount of online information, which ultimately leads to more confusion. The summary can be designed based on the type of user that is a healthcare provider or a patient. There are important sources of information on healthcare. An ontology of medical concepts, the Unified Medical Language System (UMLS), is available and can be automatically linked to terms in input articles [41].

The input for each concept can be done in several ways. Centrifuser is a summarizer that aims to help better search for information [42]. Centrifuser is an extractive domain oriented query-based MDS. The produced summary has three parts: (1) Link to query related topics for easier navigation and query reformulation. (2) A high-level overview of common parts of documents. (3) A description of the difference between recovered documents to guide people to select related items.

Medical literature on the Web is an important resource for clinicians to care for the patient [43]. A summary of medical contents will help clinicians and medical students find important and relevant information on the web faster. The presented summarization method combines various domain-specific features with other well-known features such as frequency, title, and position to improve summarization performance in the medical domain. This summarizer consists of three parts: document preprocessing, sentence ranking and summary generation. In the document preprocessing process, actions such as sentence segmentation, tokenization, stemming, and stop words removal are done. In the sentence ranking phase, some scores should be considered for the sentences. This score is calculated in terms of several factors: 1- Term frequencies. 2- Sentence similarity to the document's title. 3- Position of the sentence in the text. 4- The presence of domain related terms in the sentence. 5- The presence of new terms in the sentence. 6- Summary length.

In the summary generation phase, sentences with higher scores are selected for the summary.

Sentence ranking phase uses a knowledge base for identifying domain related terms in each sentence. This knowledge base is a list of medical terms and phrases with their weights that has been prepared by a corpus of medical news articles. In preparing the knowledge base, at first cue phrases receive a weight between 1 and 8 in terms of their impact on determining the summary worthiness and then each weight adds additional values in the range of 0 to 2 based on their position in the sentences. The position of a cue phrase in the sentence is important. A sentence that contains a cue phrase at the beginning receives a higher score than in the other positions. So a cue phrase receives a weight between 1 and 10. The knowledge base is required to identify the medical cue terms and phrases in the sentences (step (4) in sentence ranking phase). However new medical terms, such as names of genes, drugs, and diseases, are also discovered at any time. For this reason, the article introduces the idea to identify new medical terms in the text (step (5) in sentence ranking phase). An algorithm for novel medical term detection is used for this purpose. The algorithm uses two different vocabularies to determine if a term is a new medical term. One of them is a medical vocabulary and the other is an English vocabulary. If a word does not appear in the medical vocabulary, it should not be considered as a new word, because medical articles are also written using natural language, which includes words such as verbs, adverbs, and some nouns that may not be present in the medical vocabulary. Therefore, a different vocabulary is used that is made up of a corpus of natural language texts (not the medical domain). If a word was not in any of these two vocabularies, it would be considered as a novel medical term.

5- Datasets and Evaluation

5-1- Datasets

Researchers have used various datasets in the text summarization task. Many of the analyses in MDS are done on the Document Understanding Conference (DUC) datasets. This conference was organized by the National Institute of Technology (NIST) [44]. DUC has become the main international forum for discussion on text summarization [8]. The conference was first held in 2001. In each DUC edition, one or more specific tasks have been reviewed. DUC 2001 and DUC 2002 had two tasks, SDS and MDS.

Four tasks were defined for DUC 2003: 1- headline generation, which aims to produce a short 10-words summary from the single document. 2- MDS to generate summaries focusing on events in the text. 3- MDS to generate summaries focusing on viewpoints. A viewpoint is a natural language string whose length is slightly larger than a sentence and describes the important facets of the cluster. 4- MDS aims to produce short summaries of each cluster to answer a question that comes with each cluster. The requested task in DUC 2004 is to produce cross-lingual single / MDS for English and Arabic. DUC 2005 and DUC 2006 combine a set of documents to produce a brief, well-organized, fluent answer to the question. Two tasks are defined for DUC 2007. The first task is question answering, and the second task is to generate a multi-document update summary from newswire articles, assuming that the user already knows basic information about the topic. The purpose of any update summary is to give the reader new information about a specific topic. The general characteristics of the DUC data are shown in Table 4.

5-2- Evaluation

The most popular measure for evaluating summarization methods, including MDS, is ROUGE [45]. ROUGE stands for Recall Oriented Understudy for Gisting Evaluation. It measures the overlap of textual units between the system summary and the reference summary (or set of reference summaries). ROUGE evaluates the system summary from two different perspectives: recall and precision. Recall-oriented ROUGE, Precision-oriented ROUGE and F-measure-oriented ROUGE are computed by the equations (1) to (3), respectively.

$$R_r = \frac{|overlapping\ n-grams|}{|n-grams\ in\ Ref\ summary|} \tag{1}$$

$$R_p = \frac{|overlapping\ n-grams|}{|n-grams\ in\ Sys\ summary|} \tag{2}$$

$$R_f = \frac{2 \times R_p \times R_r}{R_p + R_r} \tag{3}$$

Rr: What percentage of the reference summary is covered by the system summary? This is a sign of perfection.

Rp: What percentage of the system summary is covered by the reference summary? This shows usefulness.

It is desirable that both be high, which leads to an increase in Rf. ROUGE includes a family of measures: ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S, and ROUGE-SU. ROUGE-N works on n-grams. It measures the overlap of unigrams between system and reference summaries. ROUGE-N is calculated using the equation (4):

$$ROUGE - N = \\ = \frac{\sum_{S\in\{Ref\ Sum\}} \sum_{gram_n \in S} Count_{match}\,(gram_n)}{\sum_{S\in\{Ref\ Sum\}} \sum_{gram_n \in S} Count\,(gram_n)} \tag{4}$$

ROUGE-L works with the concept of LCS. LCS is the Longest Common Substring. It calculates the overlap of LCS units between system and reference summaries. One drawback of ROUGE-L is that it does not differentiate between substrings with different distances between their elements. ROUGE-W considers the length of consecutive matches as a weight. skip-gram is any pair of words in the text with any number of gaps between them. ROUGE-S considers skip-garm overlaps between system and reference summaries. One drawback of ROUGE-S is that it does not score to a system summary that has no common skip-grams with the reference summary. That's why ROUGE-SU has combined ROUGE-S with unigrams. Between several ROUGE evaluation measures, ROUGE-1, ROUGE-2, ROUGE-Sn, and ROUGE-SUn are usually used to evaluate MDS summaries [46].

6- Challenges and Recommendations

Multi-document summarizers face more challenges than single document summarizers. For example, redundancy occurs more often in summarizing multiple documents related to a similar topic than in summarizing a single document. On the other hand, MDS performs summarization at a higher compression ratio than SDS. This makes summarizing operations more difficult. Also, summaries generated from multiple documents may be less readable than summaries generated from a single document. This is because the collection of sentences, each of which belongs to one of the documents, may result in poor coherence of the generated summary text.

There are some recommendations for each of these challenges. For example, for the information redundancy problem, before adding any candidate sentence to the summary, it is possible to calculate its overlap with the partial summary that has been produced so far. If there is a high amount of overlap, that sentence does not add to the summary. It is also possible to increase the length constraint for summaries produced from multiple documents. So there is no need to summarize at a high compression ratio. The problem of non-readability of summaries generated from multiple documents can also be reduced with techniques such as coreference resolution.

Table 4: General characteristics of DUC data [8]

Year	**DUC 2001**	**DUC 2002**	**DUC 2003**	**DUC 2004**	**DUC 2005**	**DUC 2006**	**DUC 2007**
Number of input document sets	30	59	30	50	50	50	**45**
Number of documents per set	6-16	5-15	10	10	25-50	25	**25**
Number of human summaries	3	2	4	4	4-9	4	**4**
Summary length	50/ 100/ 200/ 400	50/ 100/ 200	100	100	250	250	**250**

7- Conclusion

Searching the web for a specific topic leads to retrieving hundreds of related documents. Multi-document summarization allows the user to access the most important content of multiple text documents in a short time. In this survey we have attempted to give a comprehensive overview of the multi-document summarization techniques and domain-oriented approaches. We have categorized multi-document summarization techniques into six groups: machine learning, clustering, graph, LDA, optimization, and deep learning. A comparative overview of recent developments in each of the categories is provided. We have expressed the strengths and weaknesses of each category. Sometimes it is better to consider the domain type of the input text document, for example documents related to medicine, law, geography, etc. In this article, we also have investigated domain-oriented techniques. We have also described the most famous datasets and evaluation measures in the text summarization studies. Finally, a number of challenges and recommendations have been presented.

References

[1] G. Carenini and J. C. K. Cheung, "Extractive vs. NLG-based abstractive summarization of evaluative text: The effect of corpus controversiality," in Proceedings of the Fifth International Natural Language Generation Conference, 2008, pp. 33-41.

[2] A. Abdi, N. Idris, R. M. Alguliyev, and R. M. Aliguliyev, "Query-based multi-documents summarization using linguistic knowledge and content word expansion," Soft Computing, vol. 21, pp. 1785-1801, 2017.

[3] C. Ma, W. E. Zhang, M. Guo, H. Wang, and Q. Z. Sheng, "Multi-document Summarization via Deep Learning Techniques: A Survey," arXiv preprint arXiv:2011.04843, 2020.

[4] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz, "Multi-document summarization by sentence extraction," in Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization, 2000, pp. 40-48.

[5] R. R. K. Parchi M. Joshi, "Survey on Multi-document Summarizer," International Journal of Science and Research (IJSR), vol. 3, p. 5, 2014 2014.

[6] N. Andhale and L. Bewoor, "An overview of text summarization techniques," in Computing Communication Control and automation (ICCUBEA), 2016 International Conference on, 2016, pp. 1-7.

[7] M. Yousefiazar, "Query-oriented single-document summarization using unsupervised deep learning," 2015.

[8] M. Fuentes Fort, A flexible multitask summarizer for documents from different media, domain and language: Universitat Politècnica de Catalunya, 2008.

[9] K. Mani, I. Verma, H. Meisheri, and L. Dey, "Multi-document summarization using distributed bag-of-words model," in 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), 2018, pp. 672-675.

[10] L. Lebanoff, K. Song, and F. Liu, "Adapting the Neural Encoder-Decoder Framework from Single to Multi-document Summarization," arXiv preprint arXiv:1808.06218, 2018.

[11] S. Tabassum and E. Oliveira, "A review of recent progress in multi-document summarization," in Doctoral Symposium in Informatics Engineering, 2015.

[12] C. Shah and A. Jivani, "Literature study on multi-document text summarization techniques," in International Conference on Smart Trends for Information Technology and Computer Communications, 2016, pp. 442-451.

[13] A. Tandel, B. Modi, P. Gupta, S. Wagle, and S. Khedkar, "Multi-document text summarization-a survey," in Data Mining and Advanced Computing (SAPIENCE), International Conference on, 2016, pp. 331-334.

[14] Y. Chali, S. A. Hasan, and S. R. Joty, "A SVM-based ensemble approach to multi-document summarization," in Canadian Conference on Artificial Intelligence, 2009, pp. 199-202.

[15] S. Ma, Z.-H. Deng, and Y. Yang, "An unsupervised multi-document summarization framework based on neural document model," in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 1514-1523.

[16] P. M. Sabuna and D. B. Setyohadi, "Summarizing Indonesian text automatically by using sentence scoring and decision tree," in Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2017 2nd International conferences on, 2017, pp. 1-6.

[17] S. Ou, C. S. Khoo, and D. H. Goh, "A multi-document summarization system for sociology dissertation abstracts: design, implementation and evaluation," in International Conference on Theory and Practice of Digital Libraries, 2005, pp. 450-461.

[18] V. K. Gupta and T. J. Siddiqui, "Multi-document summarization using sentence clustering," in Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on, 2012, pp. 1-5.

[19] X. Cai and W. Li, "Ranking through clustering: An integrated approach to multi-document summarization," IEEE transactions on audio, speech, and language processing, vol. 21, pp. 1424-1433, 2013.

[20] M. Al-Dhelaan, "StarSum: A Simple Star Graph for Multi-document Summarization," in Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 715-718.

[21] A. Khan, N. Salim, W. Reafee, A. Sukprasert, and Y. J. Kumar, "A clustered semantic graph approach for multi-

document abstractive summarization," Jurnal Teknologi (Sciences & Engineering), vol. 77, pp. 61-72, 2015.

[22] G. Glavaš and J. Šnajder, "Event graphs for information retrieval and multi-document summarization," Expert systems with applications, vol. 41, pp. 6904-6916, 2014.

[23] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," the Journal of machine Learning research, vol. 3, pp. 993-1022, 2003.

[24] R. K. Roul, "Topic modeling combined with classification technique for extractive multi-document text summarization," Soft Computing, vol. 25, pp. 1113-1127, 2021.

[25] L. Na, L. Ming-xia, L. Ying, T. Xiao-jun, W. Hai-wen, and X. Peng, "Mixture of topic model for multi-document summarization," in Control and Decision Conference (2014 CCDC), The 26th Chinese, 2014, pp. 5168-5172.

[26]J. W. da Cruz Souza and A. Di Felippo, "Characterization of Temporal Complementary: Fundamentals for Multi-Document Summarization /Caracterizacao da complementaridade temporal: subsidios para sumarizacao automatica multidocumento," Alfa: Revista de Lingüística, vol. 62, pp. 121-148, 2018.

[27] A. Su, D. Su, J. M. Mulvey, and H. V. Poor, "PoBRL: Optimizing Multi-document Summarization by Blending Reinforcement Learning Policies," arXiv preprint arXiv:2105.08244, 2021.

[28] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "Multiple documents summarization based on evolutionary optimization algorithm," Expert Systems with Applications, vol. 40, pp. 1675-1689, 2013.

[29]J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach," Knowledge-Based Systems, vol. 159, pp. 1-8, 2018.

[30] A. John, P. Premjith, and M. Wilscy, "Extractive multi-document summarization using population-based multicriteria optimization," Expert Systems with Applications, vol. 86, pp. 385-397, 2017.

[31] M. Afsharizadeh, H. Ebrahimpour-Komleh, and A. Bagheri, "Automatic Text Summarization of COVID-19 Research Articles Using Recurrent Neural Networks and Coreference Resolution," Frontiers in Biomedical Technologies, vol. 7, pp. 236-248, 2020.

[32] Y. Zhang, M. J. Er, R. Zhao, and M. Pratama, "Multiview convolutional neural networks for multidocument extractive summarization," IEEE transactions on cybernetics, vol. 47, pp. 3230-3242, 2017.

[33] Z. Cao, F. Wei, L. Dong, S. Li, and M. Zhou, "Ranking with Recursive Neural Networks and Its Application to Multi-document Summarization," in AAAI, 2015, pp. 2153-2159.

[34] S.-h. Zhong, Y. Liu, B. Li, and J. Long, "Query-oriented unsupervised multi-document summarization via deep learning model," Expert Systems with Applications, vol. 42, pp. 8146-8155, 2015.

[35] S. S. Lakshmi and M. U. Rani, "Multi-document Text Summarization Using Deep Learning Algorithm with Fuzzy Logic," 2018.

[36] A. Nenkova and K. McKeown, "Automatic summarization," Foundations and Trends® in Information Retrieval, vol. 5, pp. 103-233, 2011.

[37] S. Kasundra and D. L. Kotak, "Study on Multi-document Summarization by Machine Learning Technique for Clustered Documents," 2017.

[38] Z. JIAMING, "Exploiting Textual Structures of Technical Papers for Automatic Multi-document Summarization," 2008.

[39] K. McKeown and D. R. Radev, "Generating summaries of multiple news articles," in Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, 1995, pp. 74-82.

[40] D. R. Radev, "A common theory of information fusion from multiple text sources step one: cross-document structure," in Proceedings of the 1st SIGdial workshop on Discourse and dialogue-Volume 10, 2000, pp. 74-83.

[41] O. Bodenreider, "The unified medical language system (UMLS): integrating biomedical terminology," Nucleic acids research, vol. 32, pp. D267-D270, 2004.

[42] N. Elhadad, M.-Y. Kan, J. L. Klavans, and K. R. McKeown, "Customization in a unified framework for summarizing medical literature," Artificial intelligence in medicine, vol. 33, pp. 179-198, 2005.

[43] K. Sarkar, "Using domain knowledge for text summarization in medical domain," International Journal of Recent Trends in Engineering, vol. 1, pp. 200-205, 2009.

[44] K. Hong, "Content selection in multi-document summarization," 2015.

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

[46] C.-Y. Lin, "Looking for a few good metrics: Automatic summarization evaluation-how many samples are enough?," in NTCIR, 2004.