



Automatic Text Summarization Methods: A Comprehensive Review

Grishma Sharma¹ · Deepak Sharma¹

Received: 11 March 2022 / Accepted: 9 October 2022 / Published online: 28 October 2022
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2022

Abstract

Text summarization is the process of condensing a long text into a shorter version by maintaining the key information and its meaning. Automatic text summarization can save time and helps in selecting the important and relevant sentences from the document. In extractive summarization techniques, sentences are picked up directly from the source document, whereas in abstractive summarization techniques, new sentences and phrases are generated from original source document. Majority of research is focused on extractive techniques, in the recent years, most of the research is inclined towards abstractive & hybrid text summarization methods. This paper presents comprehensive survey on various works performed for automatic text summarization methods. Detailed study is done on the Extractive & Abstractive techniques & their comparison on different aspects. The paper also discusses the research gaps & challenges that can motivate & help researchers to identify the potential areas for research in this field.

Keywords Automatic text summarization · Text summarization methods · Extractive summarization · Abstractive Summarization · Summary evaluation & Natural language processing

Introduction

In the big data era, the daily volume of text data from the variety of different sources has increased exponentially. This huge amount of text contains invaluable information & knowledge which needs to be effectively summarized. Manually summarizing this data is very challenging and difficult task for humans. Summary has many benefits like quick view of key points, easy search, selection & most important time saving [1]. Automatic text summarization system produces summary, which is compressed form of the input source document that contains a few important sentences to help the users quickly grasp the key points without reading the whole document. In the field of automatic text summarization, researchers are continuously working since 1958 [2] & trying to improve the performance of summarization techniques. Of course, there is a great improvement in this field, but still to summarize text like human, there are many

issues & challenges like readability, redundancy, coverage, & cohesion, etc. [3–5]. Text Summarization is “an open representation of representation, in diverse forms, from multiple dimensions, and through interactions in multiple spaces”. Most of the existing approaches on the basis of statistical, semantic, and linguistic, but not on the basis of understanding and representation. Text summarization research should be done with various dimensions & interactions involved in human, machine, and various other representations, including text, picture, and video [95].

Extractive and Abstractive are two major categories of text summarization systems. Extractive summarization techniques are simple, robust and select the few relevant sentences by considering their statistical features and keywords, feature scores are computed, and then, high score sentences containing these features or keywords are selected to generate the summary. Extractive summarizations gain a standard in summarization field due to its simplicity & feasibility [6], but lack in terms of sentence connectivity, readability & cohesion [7], whereas abstractive text summarization methods generates the summary from scratch. In this method, new sentences are generated by analysing the semantic information. For semantic information, abstractive summarization needs extensive knowledge of natural language processing techniques and these

✉ Grishma Sharma
grishijsharma@gmail.com
Deepak Sharma
deepaksharma@somaiya.edu

¹ Department of Computer Engineering, Somaiya Vidyavihar University, Vidyavihar, Mumbai 400077, India

techniques are more complex than extractive summarization. In the recent years with the widespread use of neural networks & deep learning models, the research attention is shifted towards abstractive and hybrid methods of text summarization. Automatic text summarization can also be classified based on usage, content of document, number of source documents, genre, output summary type, target audience, etc. Classification of text summarization is discussed in detail in "[Categorizing Automatic Summaries](#)"

This paper highlighted the current state of art & research trends in single document automatic text summarization. A concise overview of extractive & abstractive text summarization techniques, their advantages, disadvantages, comparison analysis based on datasets, and evaluation metrics is presented. Extractive summarization covered most of the techniques from statistical to machine learning. In abstractive summarization structure, semantic and more recent techniques deep learning are discussed in the detail with their pros, cons & comparison analysis based on different aspects. Sentence fusion, compression, and paraphrasing techniques for abstractive summarization are also surveyed and discussed. Important datasets, evaluation metrics, natural language tools, and challenges are also discussed in detail. Deep learning-based techniques are the current state of art in text summarization. In that different neural network-based architectures, pretrained models, sequence to sequence encoder–decoder architectures, BERT model & transformers are being used [96]. For extractive as well as abstractive different deep learning techniques are also discussed in this article. Overall, the aim is to provide the concise report about different research work done and their challenges in the automatic text summarization field which can help academics, researchers, and professionals in their research.

The article is organized as: Sect. "[Categorizing Automatic Summaries](#)" covers classification of automatic text summarization, and Sect. "[Text Summarization Approaches](#)" covers different extractive and abstractive summarization techniques and their comparison analysis. Sect. "[Text Summarization Evaluation](#)" presents evaluation metrics, Sect. "[Resources](#)" discusses the various available datasets. Sect. "[Discussion on issues / Challenges in Automatic Text Summarization](#)" covers research gaps, future directions, and finally, conclusion is drawn in Sect. "[Conclusion](#)".

Categorizing Automatic Summaries

Different methods of automatic text summarization have been proposed since 1958 [2]. Each summarization method is intended to solve a problem in a different context. As shown in Fig. 1 an automatic text summarization system can be classified using many different factors and criteria,

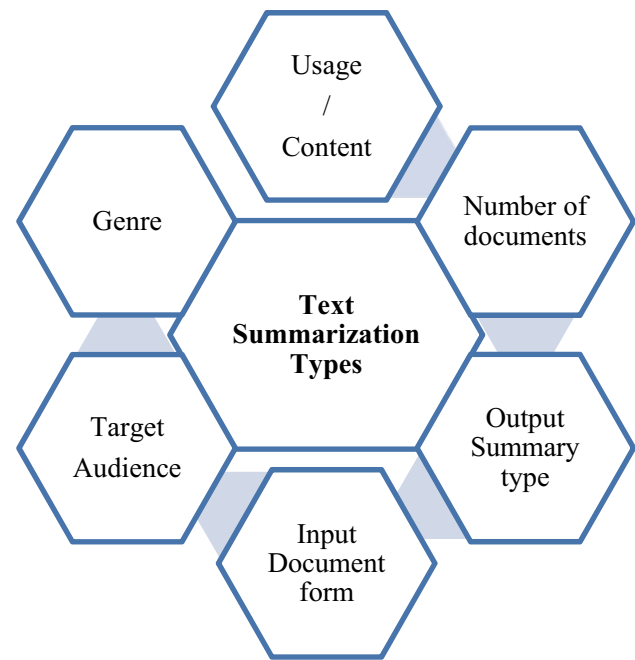


Fig. 1 Classification of automatic text summarization systems

such as usage, number of documents, genre, context, form, and output summary type [8, 9].

Usage

Based on the usage, automatic text summarization is classified as informative and indicative summary. Informative summaries convey the crucial info of the source document. Types of informative summaries are abstracts of scientific articles and synopses, whereas indicative summaries give the description of the document [10].

Number of Documents

In automatic text summarization, systems use either one input document or more than one. Single document summarizers process one input document, whereas multidocument summarizers can take many documents of the same topic [11]. In multidocument uses various different heterogeneous documents on specific topic, but redundancy is the big challenge in the multidocument summarization, because most of the information is common in documents.

Output Summary Type

Text summarization systems also can be classified basis on type of output summary generated. Output summary can either be extractive or abstractive. In extractive summary,

sentences and phrases are selected from source document by extracting the keywords, statistical features from the text, and each sentence is given a score based on these features and final summary is created by ranking these sentences [2], [2]. In abstractive text summarization methods, direct sentences are not selected from the input document, and new phrases and sentences are generated by building internal semantic representation of text with the help of natural language generation techniques. These are more coherent and grammatically correct compared to extractive summaries [13].

Input Document Form

In summarization, the form of input document significantly influences the understandability of summary. Input document can be classified basis on their form like have scale, structure & medium. Summarization methods vary based on scale of input text document. Input text document can be short text like, paragraph, tweets, and microblogs or can be full article. Structure of the document means how document is organized into different sections. In legal documents and judgements, thematic structures are used to generate summary [14]. Apart from the textual documents, summarization techniques can also be used for different file formats like images, audio, and video [15].

Target Audience

Based on target audience summarization, it can be classified as generic, query driven and update summarization. Generic Summarization gives the overall gist of the document and it is not depending on the type of input document and output summary. The Query-focused summary is based on the user query. The system picks out only the information which is related to the given query and presents a concise summary to the user. The query can be a keyword, phrase, or a sentence. Most of the search engines use this type of summarization [16]. Update summary is a special type of multidocument summary, where end user is already familiar with some facts

about the input document. Update summarization generates the new summary by neglecting the already known facts.

Genre

According to genre of document, text summarization can be classified as:

- News summary: a summary of news articles;
- Specialized: It is a summary of documents relating to a specialized domain (science, technology, law, etc.);
- Literary: A summary of narrative documents, literary texts, etc.
- Encyclopaedic: A summary of encyclopaedic documents, such as Wikipedia;
- Social networks: a summary of blogs and very short documents (such as tweets).

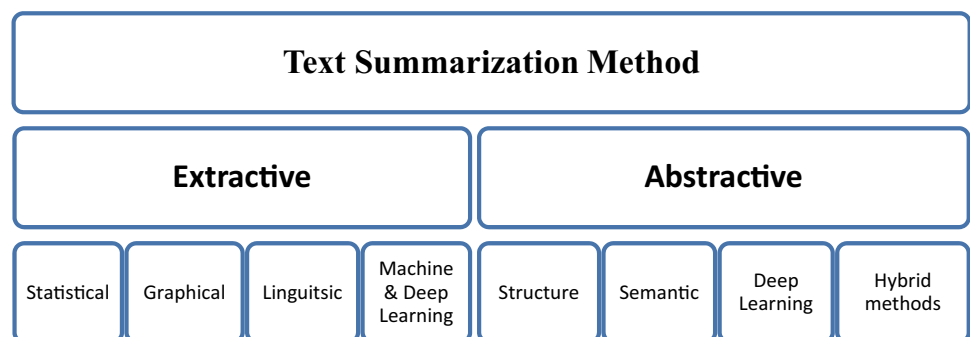
Text Summarization Approaches

In this section, different automatic text summarization methods are discussed. In this paper, the focus is given on extractive, abstractive, and neural network-based text summarization techniques. The extractive summarization produces a summary by extracting the important statistical features, and keywords from the input document. The abstractive summarization produces a summary either by rephrasing the sentences using compression and fusion techniques or by generating the new phrases, sentences through a structure, semantic or deep learning techniques. Figure 2 shows the different techniques used under the extractive as well as abstractive text summarization methods.

Extractive Summarization

Extractive text summarization methods extract the sentences from source document on the basis of different keywords and features. These features can be extracted with the help of different techniques like statistical, linguistic, graphical,

Fig. 2 Text summarization methods



and machine learning. All these techniques are discussed below in detail.

Statistical-Based Approaches

Earlier techniques of automatic text summarization are based on Statistical approaches. These methods use surface-level features like word and sentence to decide which parts of a text are important and relevant.

Word- and Sentence-Level Features

The first sentence extraction algorithm based on the term frequency to generate the summary was developed in 1958. The idea was used that, when someone writes about a given topic, certain words repeat often in the text. Thus, term relevance is considered proportional to its in-document frequency [2]. The term frequencies are later used to score and select sentences for the summary. Other good indicators of sentence relevance are the position of a sentence and have good potential to capture the importance of the sentence within the document [8]. It was demonstrated that the combination of the presence of cue words, title words, and the position of a sentence produces the most similar extracts to abstracts written by a human [17].

Latent Semantics

In natural language processing, latent semantic analysis (LSA) is a technique which is used to analyse relationships between a set of documents and the terms they contain [18]. In this technique, term–document matrix is created by m rows as document terms, and n columns as sentences. LSA learns latent features and topics by performing singular value decomposition (SVD) on term–document matrix. LSA generally used as noise and dimension reduction technique.

Graph-Based Approaches

Graph-based approaches are extensively used in text summarization, since document structures are efficiently represented by graphs. In this method documents are represented as graphs, with words, sentences, or paragraphs as nodes and edges are represented by relationship between them. Weights of the edges are based on similarities between the sentences. In graph-based approaches, the TextRank & LexRank are the two most popular methods. These two methods are the modified version of PageRank algorithm to score sentences. LexRank method [19] uses weighted cosine similarity measure to construct undirected graph; based on the similarity measure sentences are clustered into groups and the ranking of sentences is done based on their LexRank score. In TextRank [20] method is similar to LexRank except, directed

graph is constructed from the input document. In multidocument summarization, sentence to document relationship in a graph-based ranking method is also added, since in multidocument summarization, one document may be important than other documents [21].

Linguistics-Based Approaches

In text summarization using linguistics-based approaches uses natural language processing techniques like rhetoric relations, semantic, cohesion, and Coreference information to generate summaries either by extraction or abstraction.

Rhetorical Structure

Rhetorical Structure Theory (RST) is about text structure & organization that describes the relationship that exist between text elements. The hierarchical structure of a text is represented by binary tree and this structure of a text helps to identify the important parts of text to be included in the summary [22]. In [23], nested tree is constructed from source document to exploit the words and rhetorical dependency. Nested tree is composed of sentence and document tree, dependency parser is used in sentence tree where words are used as nodes which are connected by head modifier, whereas in document tree, RST is used to connect the nodes with head modifier relationships between sentences.

Coreference & Cohesion

In text summarization, co-reference & cohesion are the two very important concepts. In linguistics, co-reference resolution is the “task of finding all expressions that refer to same entity in an input text” and text cohesion is about relations between expressions which determine the text connectivity. In [24], summarization based on lexical chains was introduced, and these lexical chains can be composed with the help of different relations. Different cohesive relations like synonymy, antonymy, hypernymy, etc. between terms are determined using WordNet database. Based on the type and number of relations, chains are assigned scores and highly scored chains are selected for final summary.

Semantic Role Labelling

In text summarization, semantic representation of text plays very important role. Semantic representation can be found with the help of WordNet database and domain ontologies. In [25], one of the semantic representation semantic role labelling (SRL) is used to identify semantic roles. These semantic roles provide useful information about the text event. A text event or an action in a sentence is represented

by predicate; predicate can be the verb in the sentence. The verb in a sentence is considered as predicate, and the remaining part of the sentence can be used as the proper argument.

Machine Learning-Based Approaches

Machine Learning methods learn from the training data. Summarization task using machine learning can be done in supervised and unsupervised way. In supervised methods, with input documents, summary of each document is also provided, so that the machine can learn how to summarize, and in unsupervised methods, reference summary is not given, by extracting the features and analysing a document, machine generates the summary.

Different text document features are extracted like sentence length, frequency of words in the sentences, cue words, and position in the article etc. and combined statistically. In machine learning methods, summarization task treated as classification problem. In [26], naive-Bayes classifier is used to classify the sentences based on the extracted features. Using Bayes' rule, the probabilities of different features are learned statistically from the training data. Probability score of the sentence is calculated, and based on the score, sentence is included in the final summary.

Different classifiers use their own scoring function. Decision trees, Bayes classifier, support vector machines [27], hidden Markov models [28], and conditional random fields [29] are mostly common classifiers used in text summarization tasks. The only difference is that hidden Markov models and conditional random fields are explicitly assuming the dependency between sentences, and in other classifiers, sentences are considered as independent of each other.

Deep Learning-Based Extractive Text Summarization Approaches

With the bloom in deep learning models, neural network-based text summarization methods attracted the huge attention. The performance of these methods is much better compare to traditional methods in case of huge training data. Sentence representation and selection is the two main concerns in extractive summarization using deep learning, for sentence representation different word embedding techniques and convolutional neural networks [79], recurrent neural networks [80], and gated recurrent units [81] are used. For sentence selection, different optimization techniques are used. An unsupervised neural network-based restricted Boltzmann Machines (RBM) deep learning approach is used for extractive summarization [97]. A set of statistical and semantic features are extracted from input text documents & these features are enhanced using RBM model. This methodology gives better performance compared to TextRank, LexRank, and LSA methods.

Table 1 discusses the pros and cons of various extractive methods discussed above. Table 2 shows the comparison of various extractive text summarization methods based on technique, dataset, and evaluation metric used.

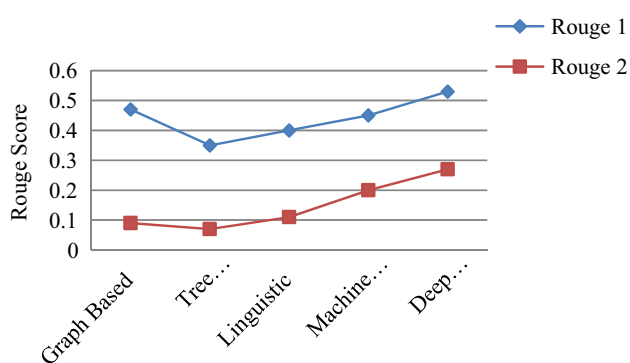
Different extractive text summarization methods are compared basis on their pros & cons, different datasets used, evaluation metrics, and their performance. According to our observation, performance of graph-based methods along with statistical methods is better in sentence scoring and selection compared to other methods. Input document length also plays very important role in performance. Although overall summary quality in terms of coherence is better in linguistic methods compared to other methods. Figure 3 shows the performance of Rouge score on DUC dataset, and from this analysis, recent deep learning methods are best among all methods on short and long document datasets.

Table 1 Advantage & disadvantages of extractive text summarization methods

Approach	Advantages	Disadvantages
Statistical	These methods require less resources, Does not require any complex linguistic tools These methods are simple and fast	Readability issues because of various features combination Redundancy in sentences: similar sentences may be included in the summary
Graph	Simple and fast Improves coherency by detecting redundant information Require less resource	More processing power Sentences similarity: may fail to identify semantically equivalent sentences [91] These methods focus only on sentence-to-sentence relationships, may not work for multi document
Linguistic	More accurate features compare to statistical methods Extracts linguistics features It can also be used for abstractive summarization	More complex resources Linguistics toolkits More processing power
Machine learning	These methods deduce rules automatically Can work on large document set Better performance with simple ML models	Lack of corpora Labelled corpus Requires a large data set of manually created extractive summaries

Table 2 Comparison between different extractive summarization methods based on method, dataset, & evaluation metric used

Reference	Method	Dataset	Evaluation metric
Kuppan S et al. [3]	Graphical approach, chunking, & ranking	DUC 2002	ROUGE1(0.53) & ROUGEL (0.49)
Steinberger et al. [18]	Latent Semantic Analysis (LSA)	Documents from Reuters datasets	Cosine & LSA similarity
Erkan et al. [19]	LexRank	DUC 2003 & 2004	ROUGE 1(~0.36–0.40)
Mihalcea et al. [20]	TextRank	Inspec database abstracts & DUC 2002	Precision, recall & F measure
Xiaojun Wan [21]	Graphical approach, multi document summarization	DUC 2001 & 2002	ROUGE1(~0.37 & ~0.39) & ROUGE2(~0.06 & ~0.08)
Daniel Marcu [22]	Rhetorical Structure Theory	40 newspaper articles from the TREC collection and five articles from Scientific American	Precision, recall & F measure
Yuta et al. [23]	Nested tree structure with rhetorical structure theory	RST- DTB Corpus	ROUGE (~0.35)
Yan S et al. [25]	Semantic role information	DUC 2006 & 2007	ROUGE1(0.41 & 0.43)& ROUGE2 (0.09 & 0.11)
Kam-Fai et al. [27]	Machine learning-based approach	DUC 2001	Precision, recall & ROUGE
John M. Conroy [28]	Hidden Markov model (HMM) machine learning-based approach	TREC data set	Precision, recall & ROUGE
Dou Shen et al. [29]	Conditional random field machine learning-based approach, sequence labelling problem	DUC 2001	Precision, recall & ROUGE
Vahed et al. [78]	Genetic algorithm based sentence extraction	DUC 2002	Precision & recall
Yin et al. [79]	CNNLM model, unsupervised, neural network-based approach, CNN language model	DUC 2002	ROUGE1 (0.51), ROUGE2 (0.27)
Cheng et al. [80]	NN-SE, supervised, convolutional neural network & recurrent neural network	DUC 2002, dailymail	ROUGE1 (0.47), ROUGE2 (0.23)
Nallapati et al. [81]	SummaRuNNer model, supervised, gated recurrent unit (GRU) neural network	DUC 2002, dailymail	ROUGE1 (0.46), ROUGE2 (0.23)

**Fig. 3** Comparison of the Extractive Text Summarization methods using Rouge1 & Rouge2 on DUC dataset

Abstractive Summarization

In abstractive summarization novel sentences are generated unlike extracting the text from the input document.

Abstractive summarization techniques require knowledge of natural language processing. It generates the summary by understanding and analysing the main concepts and key points of input document using NLP techniques. These techniques produce more concise, readable, and grammatically correct summary, and also, abstractive summarization techniques helps to achieve the more non-redundant summary compare to extractive methods by reducing the sentence size as it uses various sentence compression, generalization, and fusion techniques to merge the sentences. Abstractive summarization consists of following steps:

- i. Pre-processing
- ii. Creating an intermediate representation
- iii. Final summary generation.

In text pre-processing noise removal, tokenization, sentence segmentation, named entity recognition (NER), stop word removal, word frequency count, etc., techniques are

generally used in automatic text summarization tasks. After text pre-processing, to generate the synthetic or semantic representation of text, various different feature extraction techniques are used. Abstractive text summarization techniques broadly classified as structure, semantic, neural network, sentence compression, generalization, and sentence fusion as hybrid methods.

Structure-Based Approaches

In structure-based techniques, different form of structures is used to represent the text like template, tree, graph, ontology, and rule based. These methods generally used with other techniques like semantic, extractive, and deep learning.

In template-based methods, text feature are extracted using keywords, and to form a final coherent summary the extracted text feature snippets are populated into predefined templates. In [30], template-based technique called “GIS-TEXTER” is used to create the multidocument summaries. In this method, topic-related information is identified and converted into database entries and based on the user request sentences are added in the summary from database. Another template-based technique is used in [31] to generate the abstractive summary, noun phrases and hypernyms are used for template creation. For final summary, templates are clustered by extracting the root verbs and fused using word graph.

In tree base approaches, initially extractive summarization methods are used to extract the important text and sentences. The shallow parser is used to identify the similar sentences from the extracted text. These similar sentences are then converted into the tree structure. For final abstractive summary, predicate-argument structure or sentence fusion techniques are used. In [32], for abstractive summarization, dependency tree structure is used. In this method, phrases with common information are identified and combined using multisequence alignment, this approach is used for multi documents. Using theme selection central theme of documents is identified, then clustering algorithm is used for sentence ordering.

Mostly multidocument abstractive summarization methods utilise directly existing phrase structures extracted from input documents to generate summary. These methods can suffer from lack of coherence and consistency in merging phrases. Therefore, a novel approach for abstractive multidocument summarization through partial dependency tree extraction, recombination, and linearization is used [33]. The method generates its own topically coherent sequential structures from scratch for effective communication.

Graph data structures are widely used by many researchers in extractive as well abstractive text summarization techniques to represent the text. Directed graphs with words as nodes and the structures of sentence as edges are used in

abstractive methods. In Opinosis summarization system [34], graphs are used for generating concise summary of highly redundant opinions. The system does not require any domain knowledge and highly flexible. In this approach, initially, text representation is done using textual graph. To generate candidate abstractive summary various subpaths in the graph are explored and scored. Output summaries generated by Opinosis system have reasonable agreement with human summaries.

Mehdad et al. [35] extended the word graph method with the following novel contributions: (i) in this approach, advantage of lexical knowledge is taken to merge the similar nodes by finding their relations in WordNet; (ii) new sentences are generated through generalization and aggregation of the original ones; (iii) new ranking strategy was adapted to select the best path in the graph by taking the information content and the fluency of the sentence into consideration.

Ontology is defined as a “formal and explicit specification of a shared conceptualization”. Ontologies are defined for specific domain, and usually, they are created by domain experts [36]. Many documents on the internet are domain related, because they discuss the same concept or topic. Each domain has its own knowledge structure and that can be better represented by ontology. In these methods, sentences are reduced using reformulation and compression. Tran et al. [37] used ontologies to interpret the keyword queries, and these queries are translated into description logic (DL) conjunctive query which is evaluated with respect to underlying knowledge base.

Hennig et al. [38] described how sentences can be mapped to nodes of a flexible, wide-coverage ontology, and this mapping provides a semantic representation of the information content of sentences that improves summarization quality. For sentence classification support vector machine (SVM), hierarchical classifier is used which is trained on various sentence features. In [39], author presented a multidocument abstractive summarizer system, to capture the actual meaning and context of the document sentences, an established entity recognition and disambiguation step based on the Yago ontology is integrated into the summarization process.

Lee et al. [40] introduced Chinese news summarization, the fuzzy ontology with fuzzy concepts to model the uncertain information. The approach consists of various phases, in the first pre-processing phase, important keywords, terms & sentences extracted from news corpus. In the second phase, meaningful terms are classified on the basis of news events.

In the next phase, the fuzzy inference phase generates the membership degrees for each concept of the fuzzy ontology, and then, final summary is generated by news agent based on fuzzy ontology. Although the approach can handle the uncertain data with the help of fuzzy ontology, but it is time-consuming and limited to Chinese news.

Tanaka et al. [41] analysed broadcast news synthetically with the help of lead and body chunks of the sentences. For analysis, synthetical parsing is used. The method identifies the common phrases in the lead and body phrase, and then, insertion and substitution were applied to generate the new sentences. Substitution is applied if the body phrase has rich information and has the same corresponding phrase and if the body phrase has no counterpart insertion is applied. Insertion step basically ensures the coherency and non-redundancy and substitution enhances the information by substituting body phrase in the lead chunk.

In the rule-based methods [42], rules and categories are defined to find the important concepts from input document. In these approaches, based on input document, domain questions are formed and answers are extracted by finding the terms and concepts, and then, answers are fed into some patterns to generate the final abstractive summary.

Semantic-Based Approaches

In these methods, semantic features are extracted, and for feature extraction, text is represented by predicate-argument structure, information items, or semantic graphs, and then, this representation is given to natural language generation system to generate the final abstractive summary.

In information item-based (INIT) methods, Genest et al. [43] used “the smallest unit of coherent information in the text” called information item to find the abstractive summary. All text entities, their attributes, and different predicates between them are used as information items, and to find the information items, co-reference analysis, semantic role modelling and predicate logic analysis are used. To create the summary sentences, subject–verb–object triples information is used.

Semantic graph-based methods: These approaches are very popular and mostly used in abstractive text summarization tasks. In these methods, input text is represented as semantic relation like syntactical or ontological between the sentences. Synonymy, hyponymy, hypernymy, etc. are used as ontological relations and for syntactical relations; subject–object–verb relationship as dependency tree or syntactic tree is used. The input document is represented by graph where nodes are verbs and nouns, their semantic relationship as edges.

Moawad et al. [44] used semantic graph called “Rich Semantic Graph” (RSG) based on ontology. In this approach, author exploits a semantic graph called Rich Semantic Graph (RSG) for abstractive summary and based on ontology. The approach consists of three phases: creating a rich semantic graph for the source document, reducing the generated rich semantic graph to more abstracted graph, and finally generate the abstractive summary from the abstracted rich semantic graph. Leskovec et al. [45] presented a semantic graph

as subject–predicate–object triples along with set of other linguistic features like co-reference resolution and cross sentence pronoun resolution. Support vector machine classifier is used to extract the sentences which contain set of triplet. Abstract Meaning Representation (AMR) Graphs are also used in abstractive text summarization tasks for semantic representation. These are labelled, directed acyclic graphs and rooted graphs [46].

Semantic text representation methods aim to analyse input text using semantics of words rather than syntax or structure of text. Foland et al. [47] proposed framework for multiple documents abstractive summarization in the form of semantic representation of input documents. In this method, semantics of words are analysed by assuming text is anaphora resolved and sense disambiguated. Most significant predicate-argument structures are used for content selection. Summary is generated using a language generation tool.

To extract the predicate-argument structure from each sentence, semantic role labelling is used [48]. For assigning the sentence position numbers, SENNA semantic role labeler API is used. The similarity matrix is constructed from semantic Graph for Semantic similarity scores. After that, modified graph-based ranking algorithm is used to determine semantic similarity, predicate structure, and document set relationship. Finally, to reduce the redundancy in summarization, Maximal Marginal Relevance (MMR) is used.

Deep Learning-Based Abstractive Methods

Recently neural network-based deep learning models are being applied on various complex NLP and computer vision tasks successfully. Neural network-based text summarizers have attracted considerable attention for automatic summarization. Deep Learning is a part of machine learning that involves learning and training of data. For abstractive summarization, performance of deep learning methods is better compared to traditional methods. Most neural network-based text summarizers use the following steps for summary generation:

- i. Input document words are transformed into continuous vectors called word embeddings; these word embeddings are capable of capturing semantic similarity, relation, and context with other words in the document. Neural network-based word embedding like Word2Vec (word to vector) and GloVe (Global vectors for word representation) mostly used in various NLP applications [49, 50].
- ii. Different neural network models like “Recurrent neural networks” (RNNs) or “Convolutional neural networks” (CNNs) and “Long short-term memory”

(LSTM) are used as encoders for extracting document features.

- iii. These features are then fed to a decoder model (RNN or LSTM) for selection in extractive summarization or for generation in abstractive summarization.

In abstractive text summarization, semantic meaning representation of the whole document is captured, and final summary is generated based on this representation. In neural-based abstractive text summarization, encoder is used to represent the whole document and decoder is used to convert this representation into word sequences. In this paper, we have done survey of these methods based on encoder–decoder architecture.

To capture the meaning representation, author proposed three different kinds of encoders [52]. The first encoder is bag of word encoder which computes the summation of sentence word embeddings; this encoder does not preserve the word order. Second encoder is CNN model; in each convolution layer, sequence of feature vectors is extracted and max pool layers reduce the feature vectors by a factor of two. Third encoder is based on attention; in this encoder at each time step, document representation is produced based on previous context words generated by the decoder and feed-forward neural network-based language model is used as decoder for estimating the output probability distribution. Nallapati et al. [53] used gated recurrent unit as hierarchical attentive along with additional linguistic features like named entity recognition tags, parts of speech tags, term frequency, and inverse document frequency of the word.

In Pointer generator network model [54], single-layer bidirectional LSTM is used as encoder. Attention weights and the encoder's hidden states are used for document representation. Decoder is single-layer unidirectional LSTM. For penalizing repeated attentions on already attended words, coverage mechanism is proposed. Cohan et al. [55] proposed long document abstractive summarization model. They introduced the two new large-scale datasets of long and structured scientific papers obtained from “ArXiv” and “PubMed” papers. Scientific papers are Model which includes a hierarchical encoder for capturing the discourse structure of the document and a discourse-aware decoder for generating the summary. In this model, decoder attends to different discourse sections and allows the model to represent important information more accurately from the source. In [98], another approach for long document summarization is discussed, in that salient sentences are selected and trained using classifier than “BART—Bidirectional Auto-Regressive Transformers” pretrained model is used for abstractive summary.

Table 5 discusses the comparison of the various deep learning techniques used in the abstractive summarization based on framework, dataset, training method, and evaluation metric used.

Hybrid Methods: Sentence Compression, Fusion & Generalization

To generate abstractive summary, apart from synthetic, semantic and deep learning other approaches are hybrid methods like sentence compression, fusion, and paraphrasing. Hybrid methods are combination of extractive and abstractive. In hybrid methods, initially sentences are selected based on different extractive methods and later abstractive techniques are used to generate the summary. In hybrid methods sentence compression, fusion and generalization techniques are being used. These techniques play very important role in NLP to generate abstractive summary. In sentence compression, sentence length is reduced, while original meaning of sentence is retained. In summarization and question answering task in NLP, sentence compression addresses the problem of removing words or phrases that are not important or necessary in the generated output. Discourse or tree-based structure is mostly used in sentence compression techniques.

Galley et al. [56] treated sentence compression as optimization problem. In this approach, concept of integer linear programming (ILP) is used for inferring optimal compressions in the presence of linguistically local and global constraints. Delete- and generate-based approaches are also used in sentence compression [57]. In delete-based approaches final sentence is created by deleting the unimportant words from the sentence and by connecting the rest of the text. In generate-based models different operations like text insert, substitute and replace are used. Probabilistic approaches are used in sentence compression [58]. To compress the sentence, they used decision tree and generative noisy channel methods to compress the sentence.

Knight et al. [59] proposed model based on discriminative large margin learning framework along with compressed bigram features. For sentence compression, deep syntactic analysis and representation is done with the help of dependency parser and phrase-structure parser. In [60], hybrid approach is proposed to generate abstract summary. In this method, three-step approach is used. In the first step, sentence clusters are generated using sentence-level relationship with Markov clustering principle. In the second step, sentence ranking is done in each cluster and top weighted sentence of each cluster is fused using some linguistic rules within that cluster to generate a new sentence. In the final third step, top ranked sentences from each cluster are compressed using support vector machine classification technique to generate the abstract summary.

In [61], multidocument summarization approach called “TRIMMER” is used. It consists of a three-stage process. First, a syntactic trimmer is used to provide multiple trimmed versions of each sentence in each document of a topic set. Each of these trimmed candidates is given a

relevance score, either to a query or to the topic set as a whole. For ranking, eight different features like position, sentence relevance, and document relevance etc. Finally, sentences are chosen according to a linear combination of these features.

Generally, in sentence fusion and generalization techniques, extractive summarization methods are used to extract the important sentences and on that sentences, merging and fusion techniques are applied to generate the abstractive summary. Fusion tree is created based on the common information, extracted from the alignments of dependency structures of the sentences. Alignments can be at word, phrase, or substring level. Alignment of sentences helps to understand that how the words are related to each other. Natural language generation system is used to create the final summary from the best paths of fusion tree. [62]. A generalization technique helps in compressing the text by replacing the different concepts with one.

In [63], concept generalization & fusion approach is suggested for abstractive sentence generation. In this methodology, initially, generalizable sentences are obtained with the help of extraction and deletion method. Further sentence size is reduced by NLTK corpora and machine learning method. In [64], author introduces a novel text-to-text generation technique for synthesizing common information across documents for sentence fusion. To identify the sentences & phrases which conveying similar information, bottom-up local multisequence alignment is used. In [32], author presents a novel unsupervised sentence fusion method. German biographies were used as corpus. For a group of related sentences, dependency graph was built by aligning their dependency trees. Integer linear programming was used to compress dependency graph into dependency tree. For checking semantic compatibility of co-arguments, GermaNet and Wikipedia were used.

In [65], accurate and fast summarization model is proposed, which first selects the salient sentences and then rewrites them abstractively by compressing and paraphrasing to generate a concise overall summary. A sentence-level reinforcement learning technique is used for effectively utilizing the word-then-sentence hierarchical structure by maintaining language fluency.

Table 3 present the advantages and disadvantages of various abstractive methods discussed above. Initial most of the abstractive text summarization methods were based on template, rule, and ontology. These methods are time-consuming; require domain expertise and not scalable to different domains. Although recent deep learning methods require lot of labelled data and high end machines to process the data, but semantic structure, context and diverse domains can be easily handled with these methods.

Tables 4 and 5 show the comparison of various abstractive text summarization methods based on technique,

dataset, and evaluation metric used. According to our observation, DUC dataset is used for most of the techniques and recent deep learning methods also used CNN daily mail & Gigaword dataset. For abstractive methods, Rouge1 score is better for deep learning methods compared to other methods and performance of rouge score is shown in Fig. 4 on DUC dataset.

Text Summarization Evaluation

Text summarization evaluation is one of the most difficult tasks, because there is no ideal summary for an input document. Which is the good summary, it is very difficult to judge. Since for a same document, different humans can generate different summaries, and also, the evaluation must be accurate & fast. Because of use of various evaluation metrics and lack of a standard evaluation metric, summary evaluation task becomes very difficult and challenging [67]. Evaluation methods for text summarization can be classified as intrinsic or extrinsic [68]. Intrinsic evaluation is to evaluate the system in of it & evaluation is based on overall summary informativeness and coherence, whereas extrinsic evaluation is used to measure the impact of summary on different task-based performance, such as question answering, information retrieval & categorization, etc.

Intrinsic evaluation is done by comparing the automatic generated summary with the human written reference summary. The intrinsic methods can be of type text quality or content [69]. Text quality means linguistic aspects of the output generated summary, such as clarity, grammaticality, and coherence. The content evaluation further can be of two type co-selection or content. In co-selection, different metrics like precision, recall, & F score can be used and in content-based measures like cosine similarity, “Recall-Oriented Understudy for Gisting Evaluation” (ROUGE) & PYRAMID score can be used. From the literature survey, we found that generally content-based evaluation methods like precision, recall, F score, and ROUGE are used in summarization tasks.

For a text summarizer, input document, reference summary, and output generated summary are given, and then, we can find the true-positive, false-negative, & false-positive relations. From these relations, we can define the precision recall and f score. The “precision is the quantity of right information recovered by a system comparing to what it has recovered”, i.e., it is ratio of given in Eq. (1)

$$P = \frac{TruePositive}{TruePositive + FalsePositive} \quad (1)$$

“The recall is the quantity of right information recovered by a system comparing to what it should recover”, given in Eq. (2)

Table 3 Advantages and disadvantages of abstractive text summarization approaches

Approach	Pros	Cons
Structure	<p>Template-based methods are very informative and coherent</p> <p>In tree methods, summaries are more fluent and less redundant</p> <p>Graphical methods are simple, efficient and applicable to any domain</p> <p>Ontology-based methods produce more coherent summaries and can handle text uncertainties</p> <p>Rule-based methods generate best and high information summary</p>	<p>In template-based methods, linguistic rules, and patterns are created manually. Since template models are predefined, there is a lack of variety. The documents similarities and variations cannot be taken into account</p> <p>Tree methods ignore the context, so they miss the important phrases. More focus is on syntax rather than semantic and parser dependant</p> <p>Word graphs do not reflect the word/phrase meaning</p> <p>Ontology-based methods are time-consuming, created by expert and cannot be generalize to other domain</p>
Semantic	<p>Information item methods produce informative, non-redundant, and coherent summaries</p> <p>Multimodal-based methods compared to other semantic methods produce excellent coverage summaries</p> <p>Rich semantic graph methods produce grammatically correct & less redundant summaries</p>	<p>Rule-based methods consume time, as rules are written manually</p> <p>Information item methods require many items to fuse together as single item does not represent the complete sentence</p> <p>In multimodal methods, as document summary contains both text & images so automatic & manual evaluation, both are required</p> <p>Rich semantic graph does not support multiple documents</p>
Deep learning	<p>Encoder–decoder architecture performs very well in short text & transformers-based architecture is for long document summarization</p> <p>In deep learning, word embedding helps to capture the both semantic and syntactic structure of the text</p>	<p>Huge structured training data and more hardware resources are required to capture the distributional semantics</p> <p>RNN- & LSTM-based seq2seq encoder–decoder architectures do not capture the long-term dependency; for that attention mechanism is also added so overall design is very complex and slow training. These methods suffer from unknown tokens and repetitive words</p>
Sentence compression & fusion	<p>In sentence compression methods, original meaning of sentence is retained by compressing the length of sentence</p> <p>In sentence generalization & fusion different text concepts are replaced with one</p> <p>In sentence fusion methods, different sentence alignments can be at word, phrase or substring level</p>	<p>Highly depend on parser</p> <p>Generally algorithms have exponential space complexity</p> <p>To create grammatically correct and non-redundant summary, sentence fusion methods require dependency trees</p> <p>These methods generally require complex natural language generation (NLG) tools</p>

Table 4 Comparison between different abstractive summarization methods

Reference	Method	Dataset	Evaluation metric
Harabagiu ET AL. [30]	Template-based single & multidocument summarization	DUC—2001	Recall
Oya et al. [31]	Template-based single document summarization	AMI meeting corpus	ROUGE1(0.31) & ROUGE2(0.6)
Barzilay et al. [32]	Tree-based & sentence fusion-based multidocument summarization approach	DUC 2002	Compression ratio, Grammatical Correctness & Human evaluation
Kurisinkel et al. [33]	Tree-based multi document summarization	DUC 2004, DUC 2007 & TAC 2011	ROUGE1(0.43) score & human evaluation
Ganesan et al. [34]	Opinosis, graph-based approach	Review from Trip Advisor—Amazon	Precision, recall & F score
Yashar Mehdad et al. [35]	Word graph and sentence fusion-based approach	AMI meeting corpus	ROUGE1(0.32) score
Baralis et al. [39]	Ontology-based approach for multidocuments	DUC—2004	Maximal marginal Relevance score and human evaluation
Lee et al. [40]	Fuzzy ontology-based summarization	Chinese news articles	Precision, recall
Genest et al. [43]	Information item-based abstractive summarization	TAC 2010	Pyramid score & linguistic quality
Moawad et al. [44]	Semantic graph approach for single document summarization	“Graduate students” case study	Coherence evaluation & compression rate
Vilca et al. [82]	Semantic analysis and discourse Based	DUC 2002	ROUGE1 score (0.24) & F1 score
Sahooa et al. [83]	Rule-based approach	DUC 2002	Precision, recall, F score & ROUGE score
Liu et al. [84]	Abstract meaning representation graph-based summarization	Abstract meaning representation-AMR Bank dataset	ROUGE1 score ($P=0.51$, $R=0.40$ & $F=0.44$)
Khan et al. [85]	Semantic graph-based summarization	DUC 2002	ROUGE and Pyramid Score
Alshaina et al. [86]	Multi document abstractive summarization using predicate-argument structure	DUC 2002	Pyramid Score

$$R = \frac{TruePositive}{TruePositive + FalseNegative}. \quad (2)$$

The most used F-Score is F1-score which is a trade-off between precision & recall, given below in Eq. (3)

$$F = \frac{2 \times P \times R}{P + R}. \quad (3)$$

These evaluation measures are not useful when the system selects the other sentences than the reference sentences, so ROUGE score is generally used in abstractive summarization.

“Recall-Oriented Understudy for Gisting Evaluation” (ROUGE), proposed in [70], is a method inspired from “BiLingual Evaluation Understudy” (BLEU) score proposed by [71] used for machine translation evaluation. In these methods, the number of N grams in system output summary and reference summary is computed and from that recall value is calculated. Since in text summarization tasks,

one input document can have many reference summaries and these metrics allow the use of many reference summaries. ROUGE score has many variations like have been proposed [72]: “ROUGE-N”, “ROUGE-L”, “ROUGE-W”, and “ROUGE-S”. Equation (4) shows ROUGE-N calculations

$$ROUGE(N) = \frac{\sum_{S \in \text{sumref}} \sum_{N\text{-gram} \in S} \text{Countmatch}(N\text{-gram})}{\sum_{S \in \text{sumref}} \sum_{N\text{-gram} \in S} \text{Count}(N\text{-gram})}, \quad (4)$$

where N is N gram size, and Count (N gram) is the number of N grams in the reference summary. Count match (N gram) is the number of N grams in the candidate and reference summaries. Using these similarity scores for text summarization can present some limitations like human variation, granularity analysis, and semantic equivalence. To solve these limitations, semi-automatic approach Pyramid score is used [73].

Table 5 Comparison of abstractive summarization techniques using deep learning

Reference	Framework	Training	Dataset	Metrics
Rush et al. [51]	Attention network Bag of words Convolutional encoder Neural network language model decoder	Crossentropy, Stochastic gradient descent (SGD) optimizer	DUC & Gigaword	ROUGE1 (0.29) on Gigaword
Chopra et al. [52]	Recurrent Attentive Summarizer Convolution attentive encoder LSTM decoder	Crossentropy, stochastic gradient descent (SGD) optimizer	DUC & Gigaword	ROUGE1 (0.33) on Gigaword
Nallapati et al. [53]	Pointer generator network Hierarchical attention RNN feature-rich encoder RNN decoder	Crossentropy, Adadelata optimizer	DUC, Gigaword & CNN Daily mail	ROUGE1 (0.35) on Gigaword
See et al. [54]	Pointer generator network Coverage mechanism LSTM encoder–decoder	Crossentropy, Adadelata optimizer	CNN Daily mail	ROUGE1(0.39)
Cohan et al. [55]	Discourse Aware attention Hierarchical RNN LSTM encoder & decoder	Crossentropy, Adagrad optimizer	PubMed & arXiv	ROUGE1(0.35) on arXiv
Chen et al. [66]	Reinforce selected sentence rewriting LSTM encoder & extractor abstractor	Crossentropy + RL, stochastic gradient descent (SGD) optimizer	DUC, CNN Daily mail	ROUGE1(0.39) on CNN dailymail & Human
Iopyrev et al. [87]	Simple attention LSTM encoder–decoder	Crossentropy, RMSProp optimizer	Gigaword	BLEU
Chen et al. [88]	Distraction gated recurrent unit encoder–decoder	Crossentropy, Adadelata optimizer	CNN & LCSTS	ROUGE1(~0.27) on CNN dailymail
Paulus et al. [89]	A Deep Reinforced Model, Intra-temporal and Intra-decoder attention Weight sharing LSTM encoder–decoder	Crossentropy + RL, Adam optimizer	New York times (NYT)	ROUGE1(~0.41) & Human evaluation
Liu et al. [90]	Adversarial Training Pointer-Generator Network	Generative Adversarial Networks (GAN), Adadelata optimizer	CNN Daily mail	ROUGE1(0.39) & Human
Zhang et al. [91]	Radiology findings summarization Pointer-Generator network Background encoder	Crossentropy, Adam optimizer	Radiology Reports	ROUGE1(0.48)
Yong Zhang et al. [92]	Generative model based on convolutional seq2seq architecture Hierarchical CNN framework	Adadelata optimizer	GigaWord & CNN dailymail	ROUGE1(0.42) on CNN Dailymail

Resources

One of the major challenges in text summarization is the lack of resources, especially for an abstractive text summarization.

Nowadays, there are many powerful tools & natural language processing libraries are available for pre-processing, parsing, etc. Besides tools and libraries, annotated corpus as well can be seen as a big challenge for text summarization. In this

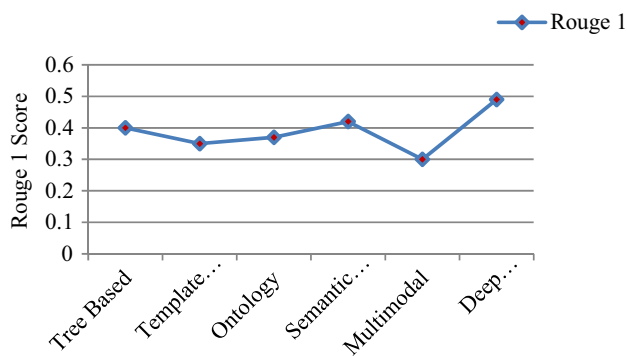


Fig. 4 Comparison of the Abstractive Text Summarization methods using Rouge1 on DUC dataset

section, some of the standard datasets available, which can be helpful to researchers, are discussed.

Document Understanding Conference (DUC) Datasets

DUC datasets are provided by the “National Institute of Standards and Technology” (NIST) and they are the most commonly used datasets in the automatic text summarization research work. The DUC conference released these datasets as a part of their summarization shared task. DUC 2001 to 2007 datasets are available on DUC website. Three different summaries are there in the each dataset, manual summary, baseline summaries, and the summaries that were generated as a challenge. In 2008, DUC became a summarization track in the Text Analysis Conference (TAC).

Gigaword Dataset

It is also popularly known as Gigaword dataset and contains nearly 10 million documents (over four billion words) of the original English Gigaword Fifth Edition [74]. It consists of articles and their headlines. It is mostly used for abstractive summarization using neural networks.

CNN/DailyMail Dataset

It is an English-language dataset containing just over 300 k unique news articles as written by journalists at CNN and the Daily Mail. The original version was created for machine reading and comprehension and abstractive question answering and the current version supports both extractive and abstractive text summarization task.

Opinosis Dataset

Opinosis dataset contains sentences extracted from user reviews on a given topic. There are total 51 topics are there with each topic having approximately 100 sentences on average. The reviews were collected from various sources like Amazon for various electronics, Tripadvisor for hotels reviews, and Edmunds for cars reviews. Dataset also contains gold standard summaries.

ArXiv Dataset

ArXiv dataset is a free, open pipeline and machine-readable repository with 1.7 million articles from arXiv, with relevant features such as article titles, authors, categories, abstracts, full-text PDFs, and more [75]. Apart from text summarization, this dataset can be used in various other applications like trend analysis, paper recommender engines, category prediction, co-citation networks, knowledge graph construction, and semantic search interfaces.

Discussion on Issues/Challenges in Automatic Text Summarization

There has been lot of research in the field of text summarization. New methods & approaches have been developed to improve the performance of existing techniques. But also there exist a lot of challenges which makes difficult the advancement in automatic text summarization, so in this section, will discuss the important issues and challenges in the field of text summarization research that needs to be addressed by the research community, a few challenges are listed below:

Dataset Availability

The large and good dataset play very important role in automatic text summarization. From the survey, we have seen most of the datasets available belong to short length of news articles like DUC, TAC. Gigawords dataset and CNN daily mail dataset are also the mostly used datasets for abstractive text summarization methods using deep learning. Recently, two new long datasets arXiv and PubMed on scientific articles are introduced in [55]. Recently, deep learning methods are proved to be very effective for abstractive text summarization, so more research and work is required to create good datasets for task & query-specific abstractive summarization. These types of datasets will be very helpful for personalized and sentiment-based summary.

Language Support

In automatic text summarization, most of the work is on English language, because it is most widely used language worldwide, and for English language, lot of natural language processing computational tools and resources like stemmers, parsing tools stop word list etc. are available. On the contrary, other multi-lingual languages have a various constraints as far as the computational tools & resources are concerned like NLP tools, spelling variations, different punctuations, etc. [76]. Therefore, to improve the quality of current summarization systems for other multi-lingual languages, there is a need to develop and improve the NLP tools and resources like Part of speech tagger, name entity recognition & parsing tools, etc.

Scalability

In text summarization, most of the work is performed on simple and compound sentences, and also, there is a need to consider the complex compound sentences, and for those, more scalable approaches are required. Summarization using deep learning algorithms requires lot of data, power and slow encoding mechanism to give good results with long documents.

Algorithm for Sentence Generalization & Fusion

In abstractive summarization sentence fusion, paraphrasing and generalization are one of the very challenging tasks. These techniques require extensive knowledge of natural language processing and not much work is done in this field. Most of the available methods are rule based and require exponential space complexity, and thus, more focus & work is required to find the algorithms in sentence generalization & fusion.

Issue of Unknown/Rare Words in Deep Learning Approaches

The problem of rare and unknown words is an important issue that can potentially affect the performance of many NLP systems, including traditional count-based and deep learning models [77]. The rare words occur less frequently in the training set and thus are difficult to learn a good representation, resulting in poor performance. It is a major challenge in text summarization methods also, due to which these methods generally fails to preserve the meaning of final summary.

Summary Evaluation

Summary evaluation is very challenge task either manually or automatically, because it is hard to define ideal or perfect summary. In text summarization assessment, mostly ROUGE score is used for evaluation. For abstractive

summary, it may be not a good metric, since these scores only matches the N grams and helps in measuring coverage of entire summary but not in coherence and non-redundancy. Therefore, there is need of an evaluation measures which can find the semantic overlap between the sentences.

Conclusion

Automatic text summarization is becoming increasingly important due to availability of vast amount of data. In this paper, survey is about the recent research and progress done in the field of automatic text summarization methods and their comparison based on different parameters. Extractive and abstractive text summarization methods are studied in detail and presented. For extractive summarization, performance of unsupervised graph-based methods is better compared to other methods on short length documents, whereas recent deep learning methods are better for large datasets and long documents. However, these summaries lack in various linguistic aspects like coherence, sentence cohesion and Coreference, whereas in abstractive summaries, they are grammatically more correct and coherent, but these methods are more difficult and challenging as they require natural language generation. Evaluation of summarization is also a challenging task; apart from automatic evaluation like Rouge score, it also requires human evaluation which is tedious and time-consuming. Finally, various challenges & gaps in existing methods are also discussed which can help researchers to identify and focus the areas where further research & improvement can be done.

Funding This work is not supported fully or partially by any funding organization or agency.

Declarations

Conflict of Interest The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Borko H, Bernier CL. Abstracting concepts and methods. London: Academic Press; 1975.
2. Luhn HP. A statistical approach to mechanized encoding and searching of literary information. *IBM J Res Dev.* 1957;1:309–17.
3. Kuppan S, Sobha L. An approach to text summarization. In: Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual (CLIAWS3), Boulder, 2009, pp. 53–60.
4. Parveen D, Mohsen M, and Michael S. Generating coherent summaries of scientific articles using coherence patterns. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Copenhagen, 2016, pp. 773–783

5. Verma P, Om H. MCRMR: Maximum coverage and relevancy with minimal redundancy-based multi-document summarization. *Expert Syst Appl.* 2019;120:43–56.
6. Gambhir M, Gupta V. Recent automatic text summarization techniques: a survey. *Artif Intell Rev.* 2017;47:1–66.
7. Yao JG, Wan X, Xiao J. Recent advances in document summarization. *Knowl Inform Syst.* 2017;53:297–336.
8. Edmondson HP. New methods in automatic extracting. *J ACM (JACM).* 1969;16:264–85.
9. Mitkov R. *The Oxford Handbook of Computational Linguistics.* Oxford: Oxford University Press; 2004.
10. Kan M-Y, McKeown K R, Klavans J.L Applying natural language generation to indicative summarization. In: *Proceedings. 8th European Workshop on Natural Lang. Generation - Volume 8 (EWNLG '01), 2001, USA.* PP, 1–9.
11. McKeown K, Radev D R. 1995 Generating summaries of multiple news articles. In *Proceedings 18th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR '95, New York.* Pp 74–82.
12. Edmondson HP. New methods in automatic extracting. *J ACM.* 1969;16:264–85.
13. Mitkov R, Unit M T. Automatic abstracting in a limited domain. In *Proceedings. Pacific Asia Conference on Formal and Computational Linguistics, 1993,* pp. 204–210.
14. Farzindar A, Lapalme G. Legal text summarization by exploration of the thematic structure and argumentative roles,” in *Text Summarization Branches Out, Barcelona.* 2004 pp 27–34.
15. Kim G, Sigal L, and Xing E P. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *Proceedings. IEEE Conference. Computer Vision and Pattern Recognition, New York, 2014,* pp. 4225–4232
16. Bhaskar P, Bandyopadhyay S. A query focused multi document automatic summarization. In *Proceedings. 24th Pacific Asia Conference on Language, Information and Computation, Sendai.* 2010 pp. 545–554.
17. Baxendale PB. Machine-made index for technical literature—an experiment. *IBM J Res Dev.* 1958;2:354–61.
18. Steinberger J, Jezek K. Using latent semantic analysis in text summarization and summary evaluation. In: *Proceedings. Seventh International Conference.* 2004 93–100.
19. Erkan G, Radev DR. Lexrank: graph-based lexical centrality as salience in text summarization. *J Artif Intell Res.* 2004;22:457–79.
20. Mihalcea R, Tarau P. Textrank: Bringing order into text. In: *Proceedings. 2004 Conference on Empirical Methods in Natural Language, Barcelona, 2004,* pp. 404–411
21. Wan X. An exploration of document impact on graph-based multi-document summarization. In: *Proceedings. 2008 Conference on Empirical Methods in Natural Language.* Processing, Honolulu, 2008, pp. 755–762
22. Marcu, D, “Improving summarization through rhetorical parsing tuning,” in *Sixth Workshop on Very Large Corpora.* 1998.
23. Kikuchi Y, Hirao T, Takamura H, Okumura M, Nagata M. Single document summarization based on nested tree structure. In: *Proceedings. 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2014 Baltimore.* 315–320.
24. R. Barzilay and M. Elhadad, 1997 “Using lexical chains for text summarization,” in *Proc. ACL/EACL'97 Workshop Intelligent Scalable Text Summarization, Madrid.* 10–17.
25. Yan S, Wan X. “SRRank: leveraging semantic roles for extractive multi-document summarization. *IEEE/ACM Transact Audio, Speech Lang Proc.* 2014;22:2048–58.
26. Kupiec J, Pedersen J, Chen F. A trainable document summarizer. In *Proceedings. 18th Annu. International ACM SIGIR Conference on Research and Development in Information Retrieval New York.* 1995. pp. 68–73.
27. Wong K.-F, Wu M, Li W. “Extractive summarization using supervised and semi-supervised learning. In: *Proceedings. 22nd International Conference on Computational Linguistics -Volume 1, Manchester.* 2008. pp. 985–992.
28. Conroy J M, O’leary D. P. Text summarization via hidden markov models. In: *Proceedings. 24th annual international ACM SIGIR conference on Research and development, New York.* 2001.Pp. 406–407.
29. Shen D, Sun J T, Yang Li Q, Chen Z. Document summarization using conditional random fields. In: *IJCAI Proceedings. 20: Twenty-Ninth International Joint Conference on Artificial Intelligence San Francisco.* 2007 2862–2867.
30. Harabagiu S. M, Lacatusu F. Generating single and multi-document summaries with gistexter. In: *Document Understanding Conferences 2002, Philadelphia,* 2002, pp. 11–12
31. Oya T, Mehdad Y, Carenini G, Ng R. A template-based abstractive meeting summarization: Leveraging summary and source text relationship. In: *Proceedings. 8th International Conference on Natural Language. Generation Conf. (INLG), Philadelphia.* 2014. pp. 45–53.
32. Barzilay R, McKeown KR. Sentence fusion for multidocument news summarization. *Comput Linguist.* 2005;31:297–328.
33. Kurisinkel L J, Zhang Y, Varma V. Abstractive multi document summarization by partial tree extraction, recombination and linearization. In: *8th International. Joint Conference Natural Language. Process.* 2017 Taipei.pp. 812–821.
34. Ganesan K, Zhai C, Han J. Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In *23rd International Conference on Computational Linguistics, Beijing.* 2010. pp. 340–348.
35. Mehdad Y, Carenini G, Tompa F, Ng R. Abstractive meeting summarization with entailment and fusion. In: *Proceedings. 14th European Workshop on Natural Language. Generation, Sofia, Bulgaria.* 2013. 136–146.
36. Mohan JM, Sunitha C, Ganesha A, Jaya A. A study on ontology based abstractive summarization. *Fourth Int Conf Recent Trends Comput Sci Eng.* 2016;87:32–7.
37. Tran T, Cimiano P, Rudolph S, Studer R. Ontology-based interpretation of keywords for semantic search. Berlin: In *The Semantic Web; 2007.* p. 523–36.
38. Hennig L, Umbrath W, Wetzker R. “An ontology-based approach to text summarization. In: *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. (Vol. 3), Washington,* 2008, pp. 291–294.
39. Baralis E, Cagliero L, Jabeen S, Fiori A, Shah S. Multi-document summarization based on the Yago ontology. *Exp Syst Appl.* 2013;40:6976–84.
40. Lee CS, Jian ZW, Huang LK. “A fuzzy ontology and its application to news summarization”, *IEEE Transact Syst. Man Cybernetics Part B (Cybernetics).* 2005;35:859–80.
41. H. Tanaka, A. Kinoshita, T. Kobayakawa, T. Kumano, and N. Kato, “Syntax-driven sentence revision for broadcast news summarization,” in *Proc. 2009 Workshop on Lang. Generation Summarisation, Suntec,* 2009, pp. 39–47.
42. Genest P E, Lapalme G. Fully abstractive approach to guided summarization. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).* 2012 354–358.
43. Genest P E, Lapalme G. Framework for abstractive summarization using text-to-text generation. In: *Proceedings. Workshop Monolingual Text-to-Text Generation, Portland.* 2011 64–73).
44. I. F. Moawad and M. Aref, “Semantic graph reduction approach for abstractive text summarization,” in *2012 Seventh Int. Conf. Computer Eng. Syst. (ICCES), Cairo,* 2012, pp. 132–138.

45. Leskovec J, Grobelnik M, Milic-Frayling N. Learning sub-structures of document semantic graphs for document summarization. In: LinkKDD Workshop, Seattle. 2004 pp. 133–138.
46. Folland W, Martin J. H. Abstract meaning representation parsing using lstm recurrent neural networks. In: Proceedings. 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver. 2017. Pp. 463–472.
47. Khan A, Salim N, Kumar YJ. A framework for multi-document abstractive summarization based on semantic role labelling. *Appl Soft Comput.* 2015;30:737–47.
48. Khan A, Salim N, Kumar Y J. Genetic semantic graph approach for multi-document abstractive summarization. In: 2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPIC), Sierre, 2015, pp. 173–181.
49. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint, [arXiv:1301.3781](https://arxiv.org/abs/1301.3781), 2013.
50. Pennington J, Socher R, "Manning C D. Glove: Global vectors for word representation. In: Proceedings. 2014 Conference on Empirical Methods in Natural Language Processing. (EMNLP), Doha, 2014, pp. 1532–1543.
51. Rush A M, Chopra S, Weston J. A neural attention model for abstractive sentence summarization. In: Proceedings. 2015 Conference on Empirical Methods in Natural Language Processing EMNLP 2015, Lisbon, 2015, pp. 379–389.
52. Chopra S, Auli M, Rush A M. Abstractive sentence summarization with attentive recurrent neural networks. In: Proceedings. 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 93–98.
53. Nallapati R, Zhou B, dos Santos C N, Gulcehre C, Xiang B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In: Proceedings. 20th SIGNLL Conference on Computational Natural Language Learning, Berlin. 2016. pp 280–290.
54. See A, Liu P. J, Manning C D. Get to the point: Summarization with pointer-generator networks. In: Proceedings. 55th Annual Meeting of the Association for Computational Linguistics, Vancouver. 2017 1073–1083.
55. Cohan A, Deroncourt F, Kim D S, Bui T, Kim S, Chang W, Goharian N. A discourse-aware attention model for abstractive summarization of long documents. In: Proceedings. 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technology. Volume 2 (Short Papers), New Orleans, 2018, pp. 615–621.
56. Galley M, McKeown K. Lexicalized Markov grammars for sentence compression. In: Human Lang. Technol. 2007: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics. Linguistics; Proc. of the Main Conf., Rochester, 2007, pp. 180–187.
57. Clarke J, Lapata M. Global inference for sentence compression: an integer linear programming approach. *J Artificial Intell Res.* 2008;31:399–429.
58. Yu N, Zhang J, Huang M, Zhu X. An operation network for abstractive sentence compression. In: Proceedings. 27th International Conference on Computational Linguistics, Santa Fe, 2018, pp. 1065–1076.
59. Knight K, Marcu D. Statistics-based summarization-step one: Sentence compression In: AAAI/IAAI Proc., 2000, pp. 703–710.
60. McDonald R. "Discriminative sentence compression with soft syntactic evidence", in 11th Conf. European Chapter Assoc. for Comput. Linguistics: Trento; 2006. p. 297–304.
61. Sahoo D, Bhoi A, Balabantaray RC. Hybrid approach to abstractive summarization. *Procedia Computer Sci.* 2018;132:1228–37.
62. Zajic D M, Dorr B, Lin L, Schwartz R. Sentence compression as a component of a multi-document summarization system. In: Proceedings. 2006 Doc. Understand. Workshop, New York, 2006.
63. E. Marsi and E. Krahmer, "Explorations in sentence fusion," in Proc. Tenth European Workshop on Natural Lang. Generation (ENLG-05), Aberdeen, 2005.
64. R. Belkebir and A. Guessoum, 2016 "Concept generalization and fusion for abstractive sentence generation. *Expert Systems with Applications*,"53, 43–56
65. Napoles C, Callison-Burch C, Ganitkevitch J, Van Durme B. Paraphrastic sentence compression with a character-based metric: Tightening without deletion. In: Proceedings. Workshop on Monolingual Text-To-Text Generation, Portland, OR, 2011, pp. 84–90
66. Y. C. Chen, and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," ArXiv preprint, arXiv: 1805.11080, May 2018.
67. Lloret E, Plaza L, Aker A. The challenging task of summary evaluation: an overview. *Lang Res Eval.* 2018;52:101–48.
68. I. Mani, "Summarization evaluation: an overview," NTCIR 2001, Tokyo, 2001.
69. Steinberger J, Jezek K. Evaluation measures for text summarization. *Comput Inform.* 2009;28:251–75.
70. Lin C. Y. Hovy E. Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings. 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, 2003, pp. 150–157.
71. Papineni K, Roukos S, Ward T, Zhu W J. Bleu: a method for automatic evaluation of machine translation. In: Proceedings. 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, 2002, pp. 311–318.
72. Lin C. Y. Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out, Barcelona, 2004, pp. 74–81.
73. Nenkova A, Passonneau R, McKeown K. The pyramid method: Incorporating human content selection variation in summarization evaluation. In: ACM Transactions on Speech and Language Processing (TSLP). 4. 4.
74. Fan J, Hoffmann R, Kalyanpur A, Riedel S, Suchanek F, Talukdar P. Proceedings of the joint workshop on automatic knowledge base construction and web-scale knowledge extraction (AKBC-WEKEX). In: Proceedings. Joint Workshop on Automatic Knowl. Base Construction and Web-scale Knowl. Extraction (AKBC-WEKEX), Montreal,
75. C. B. Clement, M. Bierbaum, K. P. O'Keefe, and A. A. Alemi, "On the use of ArXiv as a dataset," arXiv preprint, [arXiv:1905.00075](https://arxiv.org/abs/1905.00075), 2019.
76. Verma P, Pal S, Om H. "A comparative analysis on Hindi and English extractive text summarization", *ACM transactions on Asian and low-resource lang. Inf Process (TALLIP).* 2019;18:1–39.
77. C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," arXiv preprint [arXiv:1603.08148](https://arxiv.org/abs/1603.08148), 2016.
78. Qazvinian V, Hassanabadi LS, Halavati R. Summarising text with a genetic algorithm-based sentence extraction. *Int J Knowl Management Stud.* 2008;2:426–44.
79. Yin W, Pei Y. Optimizing sentence modeling and selection for document summarization. In: Twenty-Fourth International. Joint Conference. Artificial Intelligence 2015.
80. J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," arXiv preprint, [arXiv:1603.07252](https://arxiv.org/abs/1603.07252), 2016.
81. Nallapati R, Zhai F, Zhou B. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In: Proceedings. AAAI Conference. on Artificial Intelligence, San Francisco, 2017, pp. 3075–3081

82. G. C. V. Vilca and M. A. S. Cabezedo, 2017 “A study of abstractive summarization using semantic representations and discourse level information,” in *Int. Conf. Text, Speech, and Dialogue, Cham*, , pp. 482–490.
83. Sahoo D, Bhoi A, Balabantaray RC. Hybrid approach to abstractive summarization. *Proc Comput Sci*. 2018;132:1228–37.
84. F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith, “Toward abstractive summarization using semantic representations,” *arXiv preprint, arXiv:1805.10399*, 2018.
85. Khan A, Salim N, Farman H, Khan M, Jan B, Ahmad A, Ahmad I, Paul A. Abstractive text summarization based on improved semantic graph approach. *Int J Parallel Program*. 2018;46:992–1016.
86. Alshaina S, John A, Nath A G. Multi-document abstractive summarization based on predicate argument structure. In: 2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems. (SPICES), Kollam, 2017, pp. 1–6
87. K. Lopyrev, “Generating news headlines with recurrent neural networks,” *arXiv preprint, arXiv:1512.01712*, 2015.
88. Chen Q, Zhu X D, Ling Z H, Wei S, Jiang H., “Distraction-based neural networks for modeling document. In: *Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, 2016*, pp. 2754–2760.
89. R. Paulus, C. Xiong, and R. Socher, “A deep reinforced model for abstractive summarization,” *arXiv preprint, arXiv:1705.04304*, 2017.
90. Liu L, Lu Y, Yang M, Qu Q, Zhu J, Li H. Generative Adversarial Network for Abstractive Text Summarization. In: *Proc. AAAI Conference on Artificial Intelligence.*, New Orleans, 2018, pp. 8109–8110 .
91. Zhang Y, Ding DY, Qian T, Manning CD, Langlotz CP. “Learning to summarize radiology findings”, *EMNLP. Brussels*. 2018;2018:204–13.
92. Zhang Y, Li D, Wang Y, Fang Y, Xiao W. Abstract text summarization with a convolutional Seq2seq model. *Appl Sci*. 2019;9:1665.
93. Moratanch N, Chitrakala S. A survey on extractive text summarization. In: 2017 International. Conference. Comp., Commun. Signal Process. (ICCCSP), Coimbatore, 2017, pp. 1–6.
94. M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, “Text summarization techniques: a brief survey,” *arXiv preprint, arXiv:1707.02268*, 2017.
95. Zhuge, Hai. "Dimensionality on summarization." *arXiv preprint arXiv:1507.00209* (2015).
96. Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. “GSum: A General Framework for Guided Neural Abstractive Summarization”. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.
97. Sharma G, Gupta S, Sharma D. Extractive text summarization using feature-based unsupervised RBM Method. In: Agrawal DP, Nedjah N, Gupta BB, Martinez Perez G, editors. *Cyber security privacy and networking. Lecture notes in networks and systems*. Cham: Springer; 2022. https://doi.org/10.1007/978-981-16-8664-1_10.
98. Ahsaas B, Pavitra D, Kalpesh K, Pradhiksha A K, Rheeya U, Bradford Windsor, Eliot Brenner, Dominic Dotterrer, Rajarshi Das, and Andrew McCallum, “Long Document Summarization in a Low Resource Setting using Pretrained Language Models”. *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 71–80, August 5–6, 2021.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.