Computer Science • 21(2) 2020 https://doi.org/10.7494/csci.2020.21.2.3356

Sheena Kurian K.
Sheena Mathew

SURVEY OF SCIENTIFIC DOCUMENT SUMMARIZATION METHODS

Abstract *The number of research papers published each year is growing at an exponential rate, which has led to intensive research in scientific document summarization. The different methods commonly used in automatic text summarization research are discussed in this paper, along with their pros and cons. Commonly used evaluation techniques and datasets in this field are also discussed. Rouge and Pyramid scores are tabulated for easy comparison of the results of various summarization methods.*

Keywords document summarization, abstractive summarization, extractive summarization

Citation Computer Science 21(2) 2020: 141–177

1. Introduction

Scientific document summarization is the automatic process of generating a coherent summary that contains all of the important aspects discussed in the document. It is found to be as helpful as human-written summaries and more useful than the abstract written by the author himself.

A good summary should have all of the principal semantic contents mentioned in the document so that the reader can easily understand large volumes of information. By reading a good summary, the reader will be able to easily understand the current developments in the area; it should become easy for him to choose those papers that he should take for more-thorough reading and understanding. This can save lot of time and effort. It will be of special help to those who are doing research in fields requiring skills on multidisciplinary subjects.

There are two types of automatic summarization systems based on how a summary is generated: extractive and abstractive. Extractive summarization systems extract the most important sentences from the original text and forms a summary. In abstractive summarization, an understanding of the text is carried out with deep natural language analysis; then, new sentences are framed to convey the concept. For the summarization of journal articles, both extractive and abstractive methods are currently being used.

2. Extractive summarization

Extractive summarization ranks sentences and selects the highly ranked ones to be included in the summary. The ranking of sentences can be based on statistical, linguistic, or rhetorical techniques. The top-ranked sentence is included first in a summary. Then, new sentences are added to the summary only if they are considerably different from the previously included sentences. The sentences selected for summary are usually placed in the same order as they appear in the original document.

In ranking, statistical techniques are based on statistical information derived from the documents, like term frequency (tf), inverse document frequency (idf), sentence position, sentence similarity to title, sentence length, cue phrases, sentence cohesion, coverage, the occurrence of proper nouns and numbers, referring pronouns, the presence of verbs, etc. Linguistic techniques parse the document into language tokens to extract the meaning and syntax of a given sentence. Rhetorical techniques depend on the logical connections between the different parts of the text.

2.1. Extractive summarization methods

2.1.1. Surface-level and corpus-based approaches

The earliest works on scientific document summarization dates back to the late 1950s [42]. Stop words are removed from the document as a preprocessing step. Statistical measures like word frequency and relative positions within sentences are then

used to find the significant words and sentences. A writer normally repeats words that are important to the topic of discussion in a paper. The sentences are ranked based on their tf*idf scores. Highly ranked sentences are then extracted to form the summary.

A context-sensitive document-indexing model considers context and assigns greater weights to topical terms as compared to non-topical terms [20]. Content--carrying terms will have a high association with each other, and background terms will have a very low association with other terms in the document. A document summary will be centered on the topical terms in the document.

2.1.2. Clustering-based approaches

Similar sentences are clustered together to identify important themes in a document; representative sentences are then selected from each of these to form a summary [14]. This can help minimize redundancy in the summary. Each sentence can be assigned to only one cluster in this approach; however, this may not be appropriate when a sentence refers to multiple facts. Clusters are determined by the distance/dissimilarity measures between data points in a data set, the objective function that the clustering solutions aim to optimize, and the optimization procedure [67].

2.1.3. Graph-based approaches

A graph-based approach uses relationships to other sentences to rank and select sentences for summary. Graphs are built as nodes with sentences, words, or paragraphs; the relationships between them are represented by the edges. The edges are assigned weights based on similarities between sentences. Cosine similarity and tf*idf weights are commonly used. Sentences are selected to be included in a summary if the similarity index among nodes is above a defined threshold.

TextRank uses the keywords or sentences of a document to form the nodes. The edges exist between sentence nodes if they have common words and between keyword nodes if they have common nouns, adjectives, or verbs [15]. TextRank uses voting-based weighting for scoring a sentence on the basis of the incoming and outgoing edges. LexRank is an unsupervised approach that uses a cosine transform-based weighting algorithm. The importance of a sentence or keyword is found by eigenvector centrality in the graph representation of the sentences [15].

The centrality of the vertices can be calculated using PageRank algorithms. In Graphsum, a variant of a PageRank algorithm is used for multi-document summarization [4]. The lexical cohesion structure of the text can be used to determine the importance of the text [61]. The Wikipedia knowledge base is used to identify related words, and average Google normalized distance is used to calculate the similarity among nodes.

A document can be represented as a bipartite graph of a set of sentence nodes and entity nodes where there are no edges connecting the nodes within the same set [59]. The edges connect nodes from different sets. This entity graph is used to compute

the local coherence of documents and for finding the importance of sentences. The Hyperlink-induced Topic Search (HITS) algorithm is used to rank nodes in a bipartite graph. Initially, all of the entities are ranked based on similarities between the sentences and the title. The entities shared by subsequent sentences contribute to the local coherence of the text. A condensed representation of the bipartite graph (one-mode projection) is created by connecting the sentence nodes if they have common entities. The local coherence of the document is calculated as the average outdegree of the projection graph. The higher the average outdegree, the more coherent the text is. An unweighted entity graph is a directed graph with a direction that indicates sentence order.

A paragraph provides more contexts and can be chosen as a unit of extraction with more readability and coherence [51]. Paragraphs are represented as nodes in a graph and the relationships among them by edges. All pairs of paragraphs with similarities between two vectors above the threshold are connected by links, giving a text-relationship map. The text-relationship map is used to decompose the documents into segments, which are text-linked internally but largely disconnected from the adjacent text. The important paragraphs on the map are identified, and the selected nodes are traversed in text order to construct an extract or path.

Four types of paths are defined: bushy paths, depth-first paths, segmented bushy paths, and augmented bushy paths. Bushiness refers to the number of links connected to a node. A highly bushy node is related to a number of other nodes and discusses topics discussed in many other paragraphs. This is a good candidate to be included in a summary. The nodes on a bushy path are connected to a number of other paragraphs (but not necessarily to each other); so, the comprehensive coverage of the article may not be coherent and easy to read. Abrupt transition in subject matter can be eliminated by traversing the path in a depth-first search order starting with an important node and then traversing the next-most-similar node at the next step, and so on. Only the paragraphs that follow the current one in text order are candidates for the next step. Segments dealing with specialized topics have paragraphs that are not interconnected to others in the segments. A segmented bushy path selects at least one paragraph from each segment so that all aspects of the subject matter in the article are included in the summary. An augmented segmented bushy path chooses the introductory paragraph from a segment and other bushy paragraphs based on the length requirements of the summary. Introductory paragraphs are selected from each paragraph for readability.

A multi-edge-irregular graph is produced, where the number of edges in a graph between two sentences (nodes) is the number of the same words in both sentences [85]. The total number of edges is stored in a symmetric matrix. A sum vector of the rows or columns is used to rank the sentences.

An embedded graph-based sentence clustering is used for sentence ranking, where the sentences are ranked according to the mutual effects between the sentences, terms, and clusters [86]. A sentence is assigned a high rank if it is similar to many high-

ranking clusters and has many high-ranking terms. A cluster is ranked highly if it has many high-ranked sentences and many high-ranking terms. A term is ranked highly if it appears in many high-ranking sentences and clusters. The sentences that obtain high rankings are selected to be included in the summary.

Graph-based methods combine the advantages from word-frequency and sentence-clustering methods. It identifies the important content from the document but does not make any special decisions on sentence ordering. Graph-based algorithms have a limitation in that correlations among multiple terms are not considered, so some of the identified facets could be disregarded and some of the identified correlations may be misleading, as it might indicate negative associations.

2.1.4. Machine learning-based approaches (ML)

ML-based approaches learn from the data provided to the machine for summarization. There are supervised and unsupervised methods: in supervised methods, training data is provided with the summary of a document so that the machine can learn how to summarize; in unsupervised methods, the machine learns to summarize by analyzing a document.

Different features like the frequency of words, number of words common in the title, position in the article, number of sentences matching the cue words, sentence length, presence of upper-case words, etc., are statistically combined [14, 20, 57]. The sentences are selected by assigning different weights for different features and training a naive Bayes classifier [32]. A naive Bayes classifier makes the assumption that the employed features are independent of each other. Machine-learning methods have been proven to be successful in domain-specific summarization where classifiers can be trained to identify specific types of information describing the literature background.

When the edge weights in a graph are normalized to form a probability distribution so that the weight of all outgoing edges from a vertex adds up to one, the graph is a Markov chain, and the edge weights represent the probability of transition from one state to another. A hidden Markov model has fewer independence assumptions as compared to naive Bayes learners [12]. From the Markov dependencies, the probability of the next sentence being included in a summary will depend on whether the current sentence is already part of the summary. Decision trees can also be used instead of a naive Bayes classifier [35]. The features are not assumed to be independent of each other.

A challenge to machine learning is the necessity to create labeled data on which the classifiers can be trained. Human annotators selecting summary-worthy sentences can be time consuming. The automatic alignment of human abstracts and input to get labeled data of summary and non-summary sentences for learning has the issue that different writers choose different content for their abstracts.

2.1.5. Latent semantic analysis-based approaches (LSA)

LSA summarizes documents on the basis of the semantics of the text. This is an unsupervised technique that identifies text semantics based on the co-occurrence of

words [13]. The input document is represented as a word-sentence matrix. Each row corresponds to a word that appears in the input, and each column corresponds to a sentence in the input. Each entry of the matrix corresponds to the weight of a word in a sentence. LSA uses statistical correlation between a word and a passage meaning to create a similarity score between any two documents based entirely on the words they contain. LSA helps to improve cohesion (the lexical and grammatical relationships between the elements of a text) and coherence (the semantic relationships between text segments) within the summaries. LSA can identify topical shifts within the documents, which are indicators of salience and significance.

Singular value decomposition (SVD) is used to capture and model interrelationships among terms so that it can semantically cluster terms that recur in a document. SVD reduces noise and redundancy in the data by dimensionality reduction.

2.1.6. Lexical chain-based approaches

Sentences can be lexically related even if they are not adjacent to each other. A lexical chain represents a cohesive structure of the text and can be used for summarization [5]. A lexical chain is a sequence of related words in a sentence through word and synonym overlaps (one chain represents one topic). If the nouns are related to each other, a conceptual lexical chain can be found even if they are separated by many other unrelated sentences. Chains with greater lengths are more salient; they are scored on the basis of the numbers and types of relationships in the chain. Those sentences with concentrated strong chains are selected for the summary. The longest chain is chosen as the first sentence of the summary, the second longest chain is chosen next, and so on.

2.1.7. Rhetorical structure theory-based approaches (RST)

A rhetorical structure tree is a binary tree that represents the hierarchical structure of a text [45,47]. This represents the rhetorical relationships between chunks of sentences. The discourse structure of a text gives important information to identify the important spans of text to be included in the summary [40]. The smallest units of text analysis are elementary discourse units (EDUs); these are represented by terminal nodes. Adjacent EDUs are combined into larger spans through rhetorical relationships. The larger spans recursively participate in relationships that form a hierarchical tree structure covering the entire text. The discourse units in a relationship are called nuclei or satellites. A nucleus is more central to a text than a satellite. The nucleus-satellite distinction, salience, and level of an EDU in the tree are the considered properties for the summarization of a document. The satellite spans of a relationship are less essential as compared to nucleus spans. The units appearing closer to the root of RST are more important than those at lower levels. Each parent node promotes nucleus children to the parent level as they are identified as salient, and a score is given by the level it has obtained after promotion. RST can be particularly useful in the summarization of scientific articles.

An RST-based discourse tree (RST-DT) is transformed into a dependency-based discourse tree (DEP-DT). Discourse trees explicitly express the dependency relationships between textual units. The summarization procedure is considered as a Tree Knapsack problem; it can be formalized as the problem of finding the optimal rooted subtree from the discourse tree. The local properties (text cohesion) and global properties (text coherence) of the discourse structure are analyzed [23, 46]. The coherence of a summary is important in order to understand the information contained in the original document. A coherent summary can be generated by trimming the discourse tree. A discourse parser is trained for summarization, and a tree-based algorithm works with the parser to generate a summary [82].

2.1.8. Semantic information-based approaches

Semantic role labeling (SRL) identifies semantic roles like the agent, patient, and instrument of each predicate in a sentence. A predicate in a sentence represents an event or action, and semantic roles provide useful information about the event. The verb is considered to be the predicate, and the remaining part of the sentence is used as the proper arguments. Semantic information is used in SRRank to enhance the graph-based ranking algorithm for multi-document summarization [84]. SRL information is added as SRL tuple nodes in the graph. A semantic role labeler parses each sentence, recognizes the head word of an argument, and labels it. After SRL parsing, a heterogeneous graph is built for each document set that contains heterogeneous nodes (sentence nodes, SRL 2-tuple nodes, and noun nodes) and labeled directed edges (type of relationship). A node is weighted higher if it is connected to other highly weighted nodes. If a sentence is connected to other highly weighted SRL noun nodes and sentence nodes, the sentence is weighted higher. If an SRL 2-tuple is connected to other highly weighted sentence nodes, noun nodes, and SRL nodes, the sentence is weighted higher. The scores of all nodes are computed in an iterative way until convergence. Sentences of high importance and low redundancy are selected to be included in the summary.

The Simplified Lesk algorithm and WordNet are used to extract relevant sentences from a document [58]. The definitions of all meaningful words from a sentence of the document are taken from WordNet, and an intersection operation is performed between each of these meanings and the original sentence. The total number of overlaps for each sentence represents the weight of the sentence in the text. These weights represent the importance of the sentences in the text, which act as a key factor in summarization process.

2.1.9. Neural network-based approaches

A three-layered feed-forward neural network or a convolutional neural network (ConvNets) can be used in sentence extraction from labeled documents and building document classifiers [29]. A ConvNet model is divided into sentence and document levels. At the sentence level, ConvNet transforms the embedding for the words in each sentence into an embedding for the entire sentence. At the document level, ConvNet

transforms the sentence embeddings from the first level into a single embedding vector representing the entire document. The model is trained by feeding the document models to a classifier and uses backpropagation training through the sentence and document levels. Deconvolutional networks generate interpretable visualizations of the activations in the deep layers of the convolutional neural networks in the computer vision. A saliency map for the document is created by assigning an importance score to the sentence. A class prediction for the document is generated using a forward pass through the network.

2.1.10. Genetic algorithm-based approaches (GA)

Genetic algorithm-based sentence selection can be used to make a summary by the optimization principle [62]. GA avoids problems with local search techniques. A vector is defined where an element in that vector is 1 if that sentence is in the summary and 0 if it is not. This initial population of chromosomes is evaluated using a fitness function, and the best parents are selected. A new population of children is made by crossover or mutation of the parents. The summary is evaluated using a fitness function based on readability, cohesion, and topic relationship factors. The process is repeated if the stop condition is not reached [81].

2.1.11. Fuzzy logic-based approaches

A fuzzy logic-based system handles vagueness and imprecision in the data [77]. It can be integrated with statistical methods for summarization to improve the accuracy of the summarization results. Fuzzy logic can be used to measure the degree of importance and correlation to highlight the important phrases to create a summarization; it takes text characteristics like sentence length, similarity to title, similarity to keyword, etc., as the input to the fuzzy system. All of the rules needed for summarization are included in the knowledge base of the system. A value between 0 and 1 is assigned to each sentence in the output based on the sentence characteristics and rules. The value gives the degree of importance of the sentence in the summary. Fuzzy logic and wordnet synonyms are used to handle ambiguity and imprecise values [83].

Fuzzy logic system design implies the selection of fuzzy rules and membership function. The selection of fuzzy rules and membership functions affect the performance of a fuzzy logic system. A fuzzy logic system consist of a fuzzifier, inference engine, defuzzifier, and fuzzy knowledge base. In the fuzzifier, crisp inputs are translated into linguistic values using a membership function. The inference engine refers to the rule base containing fuzzy IF-THEN rules to derive linguistic values. The output linguistic variables from the inference are converted to final crisp values by the defuzzifier. Fuzzy logic requires a lot of human intervention. Fuzzy rules and classes are defined by domain experts.

2.1.12. Rough set-based approaches

Consider an information table representing the unique words in a document as different rows and the sentences as columns. Rough sets can be used for summarization by

finding data dependency and reducts, the minimal attribute subset representing the same information as the original attribute set [3]. The intersection of reducts make a core set representing the key attributes of the system that contain most of the knowledge in the data.

The relationship between objects in the information table is represented using a discernibility matrix, indiscernibility matrix, and equal-to-one matrix. A discernibility matrix contains all attributes based on which we can discern, distinguish, or differentiate between objects x and y.

An indiscernibility matrix contains all attributes based on which it is unable to discern between objects x and y. An equal-to-one matrix contains all attributes for which x and y assume a value of 1. After element absorption, no element is a proper subset of any other element. A delete operation will delete attributes as long as it is able to discern an object pair. Element absorption and element deletion are iteratively used to simplify these matrices, and the reducts thus obtained are used to select sentences for generating text summaries.

2.1.13. Classification-based approaches

Summarization can be considered to be a two-class classification problem where a sentence is labeled "correct" (value=1) if it belongs to the extractive reference summary or "incorrect" (value=0) otherwise [16]. In the testing mode, each sentence is given a value between 0 and 1. The trainable summarizer learns the patterns that identify those relevant feature values most correlated with the correct class and incorrect class. Training can be done with a genetic algorithm, mathematical regression model, feed--forward neural network, probabilistic neural network, or gaussian mixture model. These learned patterns can then assign a new sentence to either of these two classes.

The different articles are preprocessed using the NLP pipeline to obtain sentences with the best features chosen in the model-selection task [50]. The sentences with transcription factor names are retrieved and classified into structural domains (DOMs), regulated biological processes (RPs), and others. The sentences of the DOMs and RPs are concatenated for the final summary.

2.1.14. Discursive sentence compression

Highly scored sentences selected for extractive summarization may contain irrelevant information; sentence compression can produce grammatically condensed sentences that preserve the important contents [54]. The textual energy is calculated to weigh the informativeness of the segments. Each sentence is divided into elementary discourse units. Sentences with fewer informative segments are identified by statistical measures and deleted.

Discourse parsing at the document level is a significant challenge. At the sentence level, its accuracy is comparable to human performance. Discourse parsing consists of discourse segmentation, the detection of rhetorical relationships, and building a discourse tree.

2.1.15. Data summarization based on data reconstruction (DSDR)

DSDR generates a summary with sentences that can best reconstruct the original document. After stemming and stopword elimination, a document is decomposed into sentences, and a weighed-term frequency vector is created. All of the sentences form a candidate set. For each sentence in a document, DSDR selects the related sentences from the candidate set to reconstruct it by learning a reconstruction function. For the entire document, DSDR finds an optimal set of representative sentences to approximate the entire document set by minimizing the reconstruction error. Two objective functions to model the relationships among sentences are linear reconstruction (which approximates a document by linear combinations of the selected sentences) and non-negative linear reconstruction (which only allows for additive and not subtractive linear combinations). DSDR first learns a reconstruction function for each candidate sentence of an input document and then obtains an error formula by that function; then, it minimizes the reconstruction error [22].

2.1.16. Centroid-based method

Centroid represents a pseudo-document that condenses the meaningful information of a document. A centroid-based method uses a distributed representation of words in a document. Each word is represented as a vector of real numbers. A vector representation of both the centroid and each sentence of the document is projected in the vector space [68]. The sentences closer to the centroid are selected for the summary.

2.2. Challenges of extractive summarization

Extractive summarization methods have one disadvantage: all of the information in a sentence will be included in the summary once it is selected (regardless of its relevance). Unnecessary details can be removed from selected sentences by a compression method. Relevant phrases can be cut and pasted from different sentences to form a single sentence by sentence fusion. Also, textual coherence cannot be guaranteed in the case of extractive summarization if the process of anaphora resolution is not taken care of. This can be resolved with post-process NLP techniques like sentence truncation, aggregation, generalization, reference adjustment, and rewording.

Care should be taken when selecting sentences with Maximum Marginal Relevance (MMR) and minimum redundancy to be included in a summary. The selected sentences should not contain unresolved references when framing the summary. For enhancing the results of extractive summarization, the order of placement of the selected sentences is important. In some cases, it is required to do a sentence revision, fusion, or compression for better readability.

3. Abstractive summarization

Abstractive summarization methods require an understanding of natural language processing and generation as well as artificial intelligence. This is more complicated

and has been explored less than extractive methods; it requires domain specific ontology for analysis and the salience computation of concepts. However, it results in a summary that is more concise and readable with good linguistic quality.

3.1. Abstractive summarization methods

3.1.1. Structure-based approaches

Structure-based approaches can be used for summarizing news articles. It uses prior knowledge and encodes important information from documents through cognitive schemas like template-based, tree-based, ontology-based, lead and body phrase-based, rule-based, or graph-based structures [55]. Syntactic parsing is done to represent the source content as structures like predicate calculus formulas or representations such as semantic networks or dependency trees of a collection of frames. Redundant information is removed by merging graphs using the taxonomic hierarchies of the subclass relationships.

The template filling approach works only for text that is centered in a particular template [31, 55]. Language generation methods can produce a concise summary. Similar sentences are parsed using a shallow parser, and the sentences are mapped to a predicate-argument structure. The extraction rules identify text snippets that are mapped into a template slot, which are then used to generate a coherent summary. Each domain has its own knowledge structure that is better represented by ontology and can be made use of in the generation of a summary.

The lead and body phrase method is based on the insertion and substitution operations of phrases that have the same syntactic head chunk in the lead and body sentences in order to rewrite the lead sentence. The same chunks are searched in the lead and body sentences. The maximum phrases are identified and aligned using a similarity metric. A body phrase is substituted for a lead phrase if it is rich in information and the body phrase has no counterpart in the lead sentence.

The rule-based scheme uses a rule-based information-extraction module, content selection heuristics, and patterns for sentence generation. This method has the potential for creating summaries with high information density; however, these rules must be written manually, which is a very tedious and time-consuming process.

3.1.2. Semantic-based approaches

Semantic-based approaches need deeper natural language processing and natural language generation techniques for text regeneration. Semantic-based approaches use multimodal information item-based, semantic graph-based, and multimodal semantic representation models. In the information item-based approach, the summary content is generated from an abstract representation of source documents and not from the sentences in the source document [31, 37]. The syntactic subject and object are extracted, and a sentence is generated using a language generator. The frames and templates are filled with information extracted from the texts [30, 31]; this requires prior knowledge of the domain and a heavy manual effort. Template-based sentence

generation compounds the terms extracted from the text with the correct inflections. Monotony in the structure of the generated summaries is one disadvantage.

The multimodal semantic-based approach captures concepts and any relationships among the concepts. These concepts are rated based on an information density metric like completeness of attributes, the number of relationships with other concepts, and the number of expressions with the occurrence of the concept in the current document. Important concepts are selected for the summary [21].

In semantic graph-based approaches, an entire document is represented by a rich semantic graph (RSG), with verbs and nouns as graph nodes and the semantic and topological relationships between them as edges. The semantic graph is reduced to generate a concise, coherent, grammatically correct, and less redundant abstractive summary.

3.1.3. Graph-based approaches

The graph structures used in extractive summarization methods are undirected with sentences as nodes and similarities as edges, whereas it is directed with words as nodes and the structures of sentence as edges in the case of abstractive methods.

Opinosis uses graphs to produce abstractive summaries of highly redundant opinions [17]. It assumes no domain knowledge. It uses shallow NLP, leveraging the word order in the document and its inherent redundancies when generating a summary. Opinosis constructs a textual graph that represents the text to be summarized. Each node represents a word unit with positional information, and the edges represent the structure of the sentences. Each node keeps track of all sentences that have the word with a sentence identifier and its position of occurrence in that sentence. A depth-first traversal of an Opinosis graph is done to locate valid paths that represent a meaningful sentence that can be included in a summary. The Opinosis algorithm may be regarded as a word-level extractive summarization with flavors of abstractive summarization. It has elements of fusion that combine the extracted portions as well as elements of compression that squeeze out unimportant materials from a sentence.

Word graphs can be used to represent a document to produce a summary using compression and merging [34, 38]. A word graph consists of nodes that store information about words and their POS tags and edges that represent the adjacency relationships between word pairs. Two words are mapped to the same node only if they have the same parts-of-speech; however, the stop words are not mapped together. The edge weights are calculated from the frequency of occurrence of two words together in a document and from the PageRank value. A new sentence is generated from a word graph by connecting all words in the path that are directly or indirectly connected to the initial node. The initial node can be chosen as either the first word in each sentence or a word with a high tf-idf value. The shortest path will find minimum-length sentences that contain information from several sentences to include more information.

The generation algorithms may find paths among the words on the graph regardless of their syntactic correctness. This can be solved by separating the generation process into a sentence-reduction step based on input sentences, keywords and syntactic constraints, and a sentence combination step based on word graphs. The stop words, prepositions, numerals, auxiliary words, and negative words are considered to be separate nodes so that the real meaning of a sentence is not changed when new sentences are generated. Different words or phrases referring to the same concept may be represented as different nodes in the graph. Therefore, sentences with these nodes cannot be merged to create new sentences with richer information. When all nodes referring to a concept are grouped into one, a node may store multiple values with synonym words or paraphrases, and the new generated sentence may use any one of them. If the POS field has many values, the greatest value will be assigned. Three constraints are used to discard the incorrect sentences generated by the shortest path and guarantee the correctness of the new sentence: (i) the minimal length for a sentence must be three words with a subject, verb, and object; (ii) each sentence must contain a verb; and (iii) the sentence should not end with an article.

The BabelNet knowledge base is used to identify concepts and relationships from documents, and a graph is constructed. Similar concepts are extracted and placed in a related community. The sentences are rated with respect to these concepts and communities, and a summary is produced [66].

3.1.4. Fuzzy logic-based approaches

Fuzzy logic [25] can be integrated with extractive and abstractive approaches for summarization. Genetic algorithms and genetic programming can optimize the rule sets and membership functions of a fuzzy system. Both the syntactic structure of the expressions and the semantic relationships can be combined to produce summaries of good quality. The original text is pre-processed by removing redundant words, case folding, and stemming before applying as input to the summarization system. The identified syntactic parameters are term frequency-inverse sentence frequency (TF/ISF), sentence length, location of a sentence in a paragraph, similarity to the title, similarity to keywords, text-to-text coherence, integrated text-to-center, key concepts, and nouns. The semantic similarity between sentences and between word sequences in the sentences are calculated. When fuzzy logic is used, semantic similarity measures the amount by which two sentences are similar to each other rather than finding whether similar or not similar (as in conventional logic).

3.1.5. Deep learning approaches

A neural attention model where a neural language model is combined with a contextual input encoder can be used for summarization [69]. The system makes no assumption about the vocabulary of the generated summary. It can be trained directly on any document-summary pair. A convolutional attention-based encoder and a conditional recurrent neural network decoder can be used to generate a summary [7]. A graph-based attention mechanism can be used in the encoder-decoder framework [78].

Sequence-to-sequence (seq2seq) models have gained a lot of popularity in neural abstractive text summarization [74]. A hybrid Pointer-Generator network can copy words from the source text via pointing for the accurate reproduction of information and handling of out-of-vocabulary (OOV) words. Novel words can be generated through a generator. A coverage vector is also used to track and control the coverage of the source document [72].

3.1.6. Text generation approach

Text is converted to an intermediate representation based on the smallest coherent information that can be extracted from the text. It can be some property of an entity or a whole description of an event or action [18]. From each information item and the sentence from which it originates, rules are developed to generate new sentences with only one item of information. Sentences are selected to be included in a summary from the generated list of sentences by computing a score based on the frequency of items in a sentence.

The input is pre-processed, annotated, and parsed, and information items are extracted from the sentences (subject-verb-object triple). From each retrieved information item, new sentences are generated. Noun phrases (NP) are generated if a subject, object, or indirect object is present. NP generation is based on the subtree of its headword in the dependency parse tree. The head in the subtree becomes the head of the NP. Additional parts of the NP are added according to the children of the head in its parse tree. Each verb encountered forms the basis of an information item. The subject and object of the verb can be identified from the parse tree. A complement for the verb is generated if there is no object attached to it and the verb has no interesting meaning without the complement. Prepositional phrases (PP) are generated when they are the complement of a NP or when they replace the object as the complement of a verb. The head of the PP is a preposition.

Any words identified as dates or locations are not included in the sentence-generation process. Instead, they are used in the summary-generation process, with the date as a pre-modifier and location as a post-modifier. All of the generated sentences with the same date are grouped into a single coordinated sentence. If the summary length exceeds the limit, the least relevant information item is removed, and the sentence generation process is restarted; this process is repeated in a greedy way until a summary of a required length is obtained.

3.1.7. Optimized summary system

An optimized summary system identifies all research relevant novel (RRN) terms from research papers and extracts those sentences containing RRN terms [60]. RRN terms are highly correlated with the novelty of the research paper. The different categories of RRN are the main information contained in all research papers: the goal of the paper, uniqueness and difference from previous ideas (like/contrast), the research methods, research continuation/novel idea, and the research outcome. Sentences with

RRN terms are extracted and added into different predefined categories. Maximum marginal relevance (MMR) is used to reduce redundancy, and a summary is generated using data mining strategies.

3.1.8. Combining extraction with abstraction

COMPENDIUM is a text-summarization system that generates abstracts of research papers in the biomedical domain [39]. This method does not rely on specific patterns nor learn the content of the document for generating summaries. It works in two stages: first, using extractive techniques; and second, using both extractive and abstractive techniques. The extractive technique consists of surface linguistic analysis, redundancy detection, topic identification, relevance detection, and summary generation. The abstractive technique consists of word graph generation, incorrect path filtering, and information combination. Both quantitative and qualitative evaluation were done on the results. The informativeness is measured in terms of quantitative analysis and user satisfaction in terms of qualitative analysis.

3.2. Challenges of abstractive summarization

Abstractive summarization has one associated problem: the meanings of the new generated phrases may not be the same as were intended in the original document. Multi-sentence summaries sometimes fail to make sense when referring to an entity by pronoun without first introducing it. Sometimes, the generated summary may choose a less important secondary piece of information. Abstractive summarization has been mostly tested on small documents; therefore, scaling the methods to a long document is still a challenge. Abstractive methods are evaluated using the F1 scores of the Rouge-1, Rouge-2, and Rouge-L metrics. It greatly penalizes the rephrasing or rewording of words in a summary in a way that even a good summary would only have lower scores.

4. Journal article summarization

The extractive and abstractive summarization methods described in the above sections can be used for journal article summarization. The methods specifically applicable in the generation of summary of journal articles are discussed in this section. In the case of news articles, the first few sentences are always a good choice for a summary; however, this may not be so in the case of journal articles [79]. Journal articles have organized sections like introduction, problem statement, related works, methodology, results, and conclusions. The introduction section starts with general statements about the importance of the topic and history in the field; the actual contribution of the paper comes much later. The organized structure of the journal articles is used in generating summary in the methods described below.

4.1. Rhetorical structure-based approaches

Rhetorical status can be utilized in the summarization of journal articles [79]. Information about the rhetorical role of the extracted text gives the contextual information of the text. Argumentative zoning is a rhetorical classification task to classify functions of citations into seven categories: AIM specifies the research goal of the current paper; TEXTUAL specifies the section structure; OWN gives a neutral description of the methodology used in the current paper as well as the results and discussion; BACKGROUND specifies the generally accepted scientific background; CONTRAST gives statements of comparison with another work or the weaknesses of another work; BASIS has statements of agreement with another work or the continuation of another work. OTHER gives a neutral description of another researcher's work. A classifier is built to classify the sentences of a paper into these predefined categories. It uses machine learning to identify the rhetorical structure and extract sentences from the source documents by template filling to produce a summary. Sentences in AIM, BASIS, and CONTRAST can be included in the summary. From the BACKGROUND section, sentences are again classified into summary-worthy sentences and other sentences. The most appropriate sentences are chosen to form the summary.

4.2. Citation-based approaches

Citation summary is a good resource to make a summary of a target paper [63]. This model is based on analyzing other viewpoints on the contributions of the target article. The citation summary of an article is a set of citing sentences in other articles pointing to that article. Citances for a reference paper (RP) give the synopsis of its key points and key contributions within an academic community. The context of the citations determines the focus of the summary. The text surrounding the citations gives the author's viewpoints about the cited articles; if it has more than a few sentences, it is required to find a subset of them in order to provide a better and shorter summary. Although abstracts give a summarization of the document topics, they do not include metadata and critical document features.

Citation summary summarizes the reference areas in other articles related to the target paper to be summarized. The input to this summarizer consists of sentences from other papers discussing the target article. There may be a high degree of repetition in the input. Citation links between papers can be used to form good summaries of papers. These can be an overview of a research area or an impact summary of a paper using sentences from that paper itself or a citation summary formed using sentences from other papers in which that paper is cited. Citation text is an indicator of what contributions described in the target paper were more influential over time [53, 65]. Citation text may have more redundant information than abstracts, as multiple papers may describe the same contribution of a target paper. The cohesion of the citation texts of an article is consistently higher than that of its abstract. A rule-based system can identify the different reference areas in a paper where other papers are discussed and cited. These areas are classified using different cue words into (i)

an area describing the methods used in the paper, (ii) an area of discussion and comparison with a related work, and (iii) others. The text around a citation anchor can be used to assess the attitude or citation polarity (positive, negative, or neutral) of the citing paper towards the cited paper. This can be aggregated to identify the attitude of the community towards a paper.

4.2.1. Different methods in citation

Graph construction and clustering based on similarity of sentences. A network is built with citing sentences as nodes and the similarities of two sentences as edge weights [63]. Similar sentences are clustered, and a representative sentence is chosen using the graph-based method to convey information in that cluster. Clustering can be done using a support vector machine (SVM), linear regression, or neural network classifier. Similarity measures can be calculated based on tfidf, TextRank, or graph-based methods.

Two common methods for selecting sentences from clusters are cluster round-robin (C-RR) and cluster LexRank (C-LR). In C-RR, sentences are chosen from the clusters in a round-robin fashion. C-LR calculates the LexRank within each cluster and chooses the most salient sentences of each cluster. The summary of a topic can be generated from a summary of each paper and knowledge of the citation network. Different citations to an article focus on different aspects of that article [63,65]; none may cover a full description of its entire contributions. Citation summaries are more focused than abstracts and contain additional information that does not appear in abstracts. Citation networks or graphs help analyze the interplay researchers, research topics, publications, trends, venues (conference, journal), etc. [73,76]. Influence graph summarization (IGS) summarizes the citation graph to make sense of an individual's influence on the control of a citation network. Reversed citation links indicate the influence relationship between individuals: if A cites B, then B influences A.

Impact-based approach. Impact of a paper is its influence in the research of similar or related topics as reflected in the citations of the paper. The citation context and the original content of the paper is exploited to generate an impact-based summary. The sentences of the paper that have the most influential content are extracted [49]. A language model is built using the collection of all reference areas to a paper. The probability of each word occurring in a reference area is found. The importance of a sentence in the original paper is calculated. The similarity between a sentence and the language model is measured by Kullback-Leiber (KL) divergence. The importance of a sentence in the summarized article is calculated from the word probabilities estimated from the article. The final score of a sentence is a linear combination of the impact importance from the KL divergence and the intrinsic importance from the word probabilities in the input article. Impact summary is superior as measured by the ROUGE score, but is of a low linguistic quality.

Using context information. Although a citation offers a view of a cited paper, it does not consider the context of the target paper, verify the claim of the paper, nor provide

context from the reference paper. Citations to a paper may contain implicit or explicit information [64]. Implicit citations contain information about a secondary source; it contains sentences with context information. Context information is extracted to get the full meaning of the citations; this is useful for clustering similar or related papers. The fluency and readability of the summaries generated from citation contexts is high. Inconsistency between a citation summary and an article's content can be overcome by providing citation context for each citation and the discourse structure of the citing article [8]. The context will specify the assumptions and data under which the result was obtained in the referenced article. Considering the scientific discourse structure along with the citations can improve the quality of a summary [61]. The extractive summarization of citation context is done using a clustering and latent semantic analysis approach [52]. Identifying the full span of a citation context within a publication is a challenge.

Identifying keyphrases. Keyphrases can act as semantic metadata in summarization. A citation can contain zero or more nuggets or information units about a cited article. Keyphrases representing the nuggets are identified, and the best set of k sentences covering a greater number of nuggets is selected. Factoids are the different nuggets representing the same semantic unit. Different authors might use different nuggets to represent the same factoids. There may be sentences that contain references to multiple papers [1,28]. Fragments of citing sentences related to a given target reference are identified using word classification, sequence labeling, or segment classification.

Clustering on common facts. A fact is a non-overlapping contribution perceived by reading a paper. Citation sentences in one paragraph or section talk about a common fact [6]; it is represented as a set of noun phrases co-occurring in citation texts. As citations may use different terms to refer to a common fact, a term association-discovering algorithm is used to expand the terms. Citations are clustered based on the common facts that are detected. A subset of the most relevant sentences is selected to form the summary. Summaries that contain a greater number of important facts are preferred over summaries that cover fewer relevant facts [63]. The RP text spans are clustered using a single-pass clustering algorithm with a word mover's similarity score between the different RP text spans [33]. The retrieved RP spans are ranked, and the clusters are ranked according to the average TextRank scores or RP spans they contain. The most informative text span from the most informative cluster is selected for the summary in order until a certain summary length is reached.

Citation function. The citation function is the author's reason for citing a given paper [80]. Citations in a running text are divided into four classes: (i) conceptual or operational use; (ii) evolutionary or juxtapositional; (iii) organic or perfunctory; and (iv) confirmative or negational. It is seen that 40% of the citations are perfunctory – done out of politeness, policy, or piety. It is important to know whether a certain article criticizes another and what the criticism is or if the current work is based on a prior work.

Vector space and language model. Machine learning estimates the relative importance of the sections of the document as well as the citing sentences [11]. This consists of data pre-processing and segmentation, term selection, latent term weight estimation using a vector space model or bigram language model, and sentence selection. Individual sections of the paper are isolated and extracted, and stemmed word bigrams with high mutual information is formed. A vector space model is based on the term frequency matrix representation of the document and a nonnegative matrix factorization approximation for rank reduction. A bigram language model is built on selected bigrams from each document selection.

4.2.2. Analysis of citation-based approaches

Analysis of a summary generated with a citation summary network is done using fact distribution and similarity measures. In fact distribution, annotators read the citation summary of each paper and extract a list of non-overlapping contributions of that paper. A manually created union of shared and similar facts are found. The fact-distribution matrix of an article has rows that represent sentences in the citation summary and columns that represent facts. If a sentence has a particular fact, the corresponding entry in the matrix will be 1. A network is built with citing sentences as nodes and the similarities of two sentences as edge weights. The similarity value will be high for pairs of sentences that cover the same fact [63].

4.3. Deep learning methods

Many journals require authors to submit highlight statements along with their manuscripts; these can be used to improve summarization performance [10]. Each sentence is encoded as mean averaged word embeddings and as recurrent neural network encoding. The AbstractRouge score, location, numeric count, title score, keyphrase score, tfidf, document tfidf, and sentence length are the eight features used for training a neural network. Seven different models are created for comparison using four inputs named as: S, the sentence encoded with an RNN; A, the vector representation of the abstract of a paper created by averaging the word vectors of every non-stopword word in the abstract; F, the eight features listed above; and Word2vec, the sentence represented by taking the average of every non-stopword word vector in the sentence. Model names containg 'Net' use a neural network with one or multiple hidden layers. The different models created are single feature models; FNet; word2vec and word2vecAF; SNet; SFNet; SAFNet; (SAF+F) and (S+F) ensemblers, and are compared with exisitng models. The SAF+F ensemble gave the best score when compared to LexRank, TextRank, KLSum, LSA, and SumBasic.

4.4. Unsupervised approaches

Each section in the document is represented as a vector, with words or expressions being the features of this vector. Feature maximization detects the specific features that describe each section and discriminate it from the others [71]. The Feature F-measure

of each word is calculated after removing the stop words. The weights of each sentence are calculated by aggregating the weights of the words. Summaries are then produced with the highest-weighted sentences.

5. Evaluation metrics

The evaluation methods range from purely manual approaches to purely automatic ones. In the manual approach, a human evaluates a candidate summary from different points of view like coverage, grammaticality, or style. A high score is given to a candidate summary based on grammaticality, non-redundancy, focus, structure, and coherence. Automatic approaches compare segments of texts from the candidate summary with several of the reference abstracts.

There are intrinsic and extrinsic methods for evaluating a summary [24]. Intrinsic methods evaluate the quality of a summary by (i) a comparison to a gold standard using co-selection measures like precision, recall, F-scores, and relative utility and content-based measures like cosine similarity, unit overlap, longest-common subsequence (LCS), Recall-Oriented Understudy for Gisting Evaluation(ROUGE), PYRAMID, and Latent Semantic Analysis-based (LSA) measures, and (ii) a text quality evaluation of a summary produced according to some scale like readability, fluency, coverage, structure and coherence, referential clarity, etc. Extrinsic evaluation measures the impact of a summary on task-based performance like document categorization, information retrieval, question answering, etc.

Precision measures how many of a system's extracted sentences were good. Recall measures how many good sentences the system missed. F-score is a composite measure that combines precision and recall. Relative utility gives the confidence values of inclusion of sentences in a summary. Rouge measures are evaluation metrics for fixed-length summaries based on n-gram overlaps between the terms and phrases in the sentences [36]. It compares the quality of a computer-generated summary versus a human-generated summary. The different Rouge measures commonly used are: Rouge-N, the n-gram overlap between the system summary and reference summary; Rouge-L, the longest common subsequence; Rouge-S, the skip-bigram coocurrence; and Rouge-SU the extension of Rouge-S with unigram as counting unit. Rouge measure is used in the Text Analysis Conference (TAC), a series of workshops for evaluating research in NLP. The effectiveness of Rouge was promising on news-summarization data. It is not that effective in evaluating scientific summaries, as it differs from news data in factors like length, complexity, and structure [9]. It turns out to be inefficient if the summary is generated using reformulation techniques with their own wordings.

Pyramid scores are also compared for journal summarization methods. Pyramid is a method of evaluation used at the sentence level to overcome these issues. Pyramid represents a gold standard summary for an input set of documents against which the automatically generated summaries are compared [56]. The facts to be included in a summary are called summarization content units (SCU). SCUs are given a score from 0 to 1. SCUs appearing in more human summaries are weighted more heavily.

A tier contains all (and only) SCUs with the same weight. Each fact falls within a tier of the pyramid. Each tier shows the number of sentences in which a fact appears. There will be many SCUs with low scores at the base of the pyramid and a few SCUs with high scores at its top. The number of tiers in a pyramid is equal to the citation summary size [63]. Pyramid evaluation identifies all of the SCUs contained in a candidate summary and calculates a score based on the number and weight of the SCUs it contains. This is more precise than Rouge, but it requires manual work to identify the SCU and calculate the weights necessary for evaluation.

In LSA-based evaluation, the most important topics in a document that can be used in summary content evaluation are found [75]. LSA evaluates summary quality by finding content similarities between a reference document and the summary.

Automatic methods based on comparison with manual reference. When manually created citation text summaries are used as references, summaries generated from citation texts have better rouge scores than summaries generated from abstracts and full papers. Summaries from abstracts performed better than full papers. When manually created, abstract summaries are used as references; summaries generated from abstracts obtained better rouge scores than summaries generated from citation texts and full papers. Summaries generated from citation texts performed better than full papers. This shows that abstracts and citation texts are richer in survey-worthy information.

Automatic evaluation with no manual reference. Reference summaries are costly to produce for use in automatic evaluation. The definition and weighting of SCUs is also a difficult and time-consuming task. The input text is used to produce a summary containing valuable information to evaluate the summary derived from it [41]. Four classes of features can capture aspects of input text: input and output text coverage, information theoretic properties of the distribution of words in the input, the presence of topic words, and similarities between documents in multi-document inputs. A set of candidate summaries was compared using these features and the Jenson-Shannon (JS) measure. This method does not perform well for biographic information or for the summarization of opinions. JS divergence is a metric for evaluating summaries without the need for human-generated summaries. It is correlated with the intrinsic quality of the summary.

6. Community efforts in text summarization

TIPSTER was the first large-scale developer independent evaluation of automatic text summarization systems. TIPSTER text summarization evaluation (SUMMAC) focuses on evaluating summaries based on (i) determining a document's relevance to a topic for query-relevant summaries, (ii) determining categorization for generic summaries, and (iii) establishing whether summaries can answer a specified set of questions by comparison to a human-generated model summary [44]. Summaries were rated in terms of confidence in decision, intelligibility, and length. SUMMAC evaluation judges summarization systems in terms of their usefulness in specific

summarization tasks and to gain a better understanding of the issues involved in building and evaluating them.

CL-SciSumm Pilot Task 2014 was conducted as part of BioMed Summ Track at the Text Analysis Conference (TAC) to encourage community efforts in the field of "faceted summaries" in the domain of computational linguistics. It was comprised of annotated citation information connecting research papers with citing papers. Task 1A was to identify the text span in the reference paper that corresponds to citances from the citation paper. This was scored using word overlaps with the gold standard measured by the ROUGE-L score. Task 1B was to identify the discourse facet for each cited span from a predefined set of facets. It was scored using precision, recall, and F1. Task 2 (optional) was to generate summary. It was evaluated against the abstract using ROUGE-L [26].

CL-SciSumm 2016 Shared Task is the first medium-scale shared task on scientific document summarization in the CL domain. It was organized as part of the BIRNDL workshop at JCDL 2016. The third CL-SciSumm Shared Task in 2017 provides resources to encourage research in entity mining, relationship extraction, question answering, and other NLP tasks for scientific papers. The data set of the fourth CL-Scisumm Shared Task 2018 was extended to 60 annotated sets of citing and reference papers in the CL domain and the three types of summaries (abstract by author, community summary collated from reference spans of its citances, and human-written summary by annotators) for each reference paper [27].

6.1. Methods for identifying reference paper (RP) text spans (Task1A)

Pairs of reference and citance sentences can be modeled as a feature vector and are used to train binary classification algorithms to detect whether there is a match [70]. Position-related features are also used here. The problem can be framed as an information retrieval (IR) task where the RP sentences are ranked according to their relevance of the citance; the highly ranked sentences are selected [33]. The output can be extended with RP sentences adjacent to the top-ranked sentence if they come in k top-ranked sentences. The rankings can be based on

- Semantic similarity where an RP sentence can be semantically similar to the citance with little or no lexical overlap, aggregate sentence embedding similarity, linear discriminant analysis, or word mover's similarity.
- Lexical similarity identified by finding cosine similarity, unigram overlap or Bag of Words (BOW) similarity, or vector space model between citance and RP text's sparse tfidf vectors.
- Word embedding-based similarity measure, CNN over word embeddings.
- Sentence similarity methods using Siamese Deep learning networks and the positional language model approach.
- Entity overlap computed as a Jaccard coefficient over linked entities from the citance and RP sentence, BabelNet synset embeddings cosine similarity.

- Positional features found from the relative section positions and sentence positions of a candidate RP sentence in the RP and citance in a citation paper (CP).
- Frequency-based features
- Graph-based ranking algorithm where vertices in a graph are ranked based on their importance given by connectedness within the graph.
- SVM rank with lexical and document structural features to rank reference text sentences for each citance.
- Lexical and syntactic cues to identify reference scope of citance. The stop words are removed, and BOW are generated for statements in RP and cited text. Each word is stemmed to find frequent unigrams or bigrams. The distance between bigrams is computed in statements in RP. Parse the cited text to identify dependency overlap between the cited text in CP and RP by matching dependency tags and the position of words.
- Feature Maximization with cosine similarity. The significance of words of a reference paper is evaluated by extracting its feature F-measure; the harmonic mean of feature recall and feature predominance. The sentences of the reference paper and the citations (query) are represented as vectors. Cosine similarity is applied to select the most relevant sentence from the reference paper according to the query [71].
- Binary classification problem with different feature sets. Logistic regression with content-based features derived on topic and word similarities gave good results.
- Random forest model and voting strategy to select the most related text spans.
- Weighted voting-based ensemble of classifiers like linear SVM, SVM using a radial basis function kernel, decision tree, and logistic regression to identity the reference span.

The different methods were evaluated as overlaps in sentence ID marked against a gold standard created by human annotators. The number of overlapping sentences was used to calculate precision, recall, and F1 scores. Rouge-2 and Rouge SU4 scores were also calculated. The average performance on every task was obtained as the average performance on each of three independent sets of annotations as well as the three summaries [26, 27, 33, 70].

6.2. Methods for identifying reference paper facet (Task 1B)

Each pair of reference and citance sentences is classified to the five predefined facets as Aim, Hypothesis, Implication, Results, and Method. The different methods used for classification are:

- Binary classifiers trained for each label using a Support Vector Machine (SVM) and Convolutional Neural Network (CNN). SVM classifiers gave good scores over CNN. CNN was found to be successful for a range of short-text classification tasks. The limited size of the training data may be a reason for the poor performance of CNN classifiers as compared to SVM [33].
- Multi-class classifiers trained with a feature vector for each sentence constructed as the average of word embeddings for the terms in the sentence.

- Random forest classifier with section features, tfidf features, and sentence position features.
- Linear regression implementation of weka used along with the GATE system [70].
- Rule-based systems and supervised machine-learning algorithms like Decision tree classifiers and the K-nearest neighbor classifier.
- A similarity function to select the best-matching reference span with the same facet as the citance [26]. A co-occurrence graph can be constructed to highlight words that are not in the query but that should be taken into consideration in the selection process. This method has the advantage that it does not require any training test nor parameter learning.

The methods are evaluated as a proportion of the correctly classified discourse facets by the system. As it is a multi-label classification, the task was also scored based on precision, recall, and F1 scores [26, 27, 33, 70].

6.3. Summarize reference span (Task 2)

The methods used in summarizing the reference span are listed below.

- Cluster RP segments retrieved for individual citances according to their semantic textual similarity and select the most informative sentence from each cluster according to the TextRank Score [33].
- Maximum Marginal Relevance (MMR) to choose sentences ensuring that new information is actively being added to the summary [41].
- Query focused summarization task with citances as queries.
- Word Movers Distance similarity to construct new kernel matrix used in Determinantal Point Processes (DPPs).
- Group sentences into three clusters (motivation, approach, and conclusion) and then extracted sentences from each cluster to combine into a summary.
- Select sentences from RP that are most relevant to the CPs using various features. CNN can be used to learn the relationship between a sentence and a scoring value can indicate its relevance.

The methods are evaluated using ROUGE N scores between the system output and the gold standard summaries. It is seen that the lexical methods worked well with the structural and semantic characteristics that were unique to scientific documents and was complemented with domain specific word embeddings in a deep-learning framework [26, 27, 33, 70].

7. Commonly used data sets

- DUC datasets released in association with the international conference for performance evaluation in the area of text summarization of single document.
- Medline PubMed dataset providing keyword searches for retrieving abstracts.
- Text Analysis Conference (TAC) dataset for the evaluation of areas in NLP (like the summarization of multiple documents).

- Computational Linguistics Scientific Document Summarization Shared Task Corpus (CL-Scisumm) with abstract, community, and human summary.
- TIPSTER Text Summarization Evaluation Conference (SUMMAC) with xml markup.
- ACL Anthology Network corpus (AAN).
- Topic Detection and Tracking (TDT).
- Many evaluation competitions like TREC, DUC, and MUC have created data sets for training and have established baselines for performance levels. However, the strategy for evaluation is still not universal.

8. Results of summarization methods

The Precision, Recall, and F-measure values of different Rouge measures are compared for extractive, abstractive, and journal summarization methods.

8.1. Comparison of extractive summarization methods

The results of various extractive summarization methods are given in Table 1. It can be seen that statistical-based methods like fuzzy, genetic, and graph-based systems have better Rouge scores than the classification-based, clustering-based, centroid-based, semantic role information-based, and discourse-based methods. Hence, these methods can be effectively used in extractive summarization.

Table 1
Comparison of Extractive Summarization Methods (Precision, Recall, F-Measure, and Average of Rouge1, Rouge2, and Rouge-SU4 evaluation metrics)

Type	Method	Dataset	Rouge-1				Rouge-2				Rouge-SU4			
			P	R	F	A	P	R	F	A	P	R	F	A
Clustering--based	Evolutionary algorithm on clustering and extracting [67]	DUC 2001	–	.459	–	–	–	.193	–	–	–	.218	–	–
		DUC 2002	–	.454	–	–	–	.189	–	–	–	.213	–	–
Graph--based	Mutual reinforcement using embedded graph-based sentence clustering [86]	DUC 2001	–	.319	–	–	–	.068	–	–	–	–	–	–
	Graph Sum [4]	DUC 2004	–	–	–	–	.099	.093	.097	–	.021	.015	.019	–

Table 1 (cont.)

Type	Method	Dataset	Rouge-1				Rouge-2				Rouge-SU4			
			P	R	F	A	P	R	F	A	P	R	F	A
Graph--based	Text segmentation using wikipedia [61]	DUC 2001	–	–	.487	.480	–	–	–	.189	–	–	–	–
		DUC 2002	–	–	.483	.471	–	–	–	.129	–	–	–	–
	Multi-edge irregular graph [85]	Dataset1	–	–	–	–	.089	.502	.151	–	–	–	–	–
		Dataset2	–	–	–	–	.075	.652	.135	–	–	–	–	–
	Integrating importance, non-redundancy and coherence [59]	DUC 2002	–	–	–	.485	–	–	–	.23	–	–	–	254
		PLOS-against editor summary	–	–	–	–	–	–	–	.098	–	–	–	.131
		PLOS-against author abstract	–	–	–	–	–	–	–	.189	–	–	–	.224
Genetic algorithm	Genetic algorithm--based sentence extraction [62]	DUC 2002 Length= 400	.496	.378	–	–	–	–	–	–	–	–	–	–
		Length= 200:	.459	.428	–	–	–	–	–	–	–	–	–	–
Semantic role information	SRRank: Leveraging Semantic roles [84]	DUC 2006	–	.413	–	–	–	.090	–	–	–	.149	–	–
		DUC 2007	–	.430	–	–	–	.119	–	–	–	.171	–	–
Discourse--based	Trimming the discourse Tree [23]	RST-DT -against abstract without stopword	–	–	–	.321	–	–	–	.112	–	–	–	–
		-against abstract with stopword	–	–	–	.346	–	–	–	.107	–	–	–	–
		-against extract without stopword	–	.305	–	.451	–	–	–	.324	–	–	–	–
		-against extract with stopword	–	–	–	.501	–	–	–	.337	–	–	–	–

Table 1 (cont.)

	Task Oriented Discourse Parsing [82]	RST-DT	–	.330	.380	–		.116	.151	–	–	–	–	–
		DUC 2001	–	.358	.373	–		.121	.130	–	–	–	–	–
Statistical methods	GA, MR, FFNN, PNN, GMM-based models [16]	DUC 2001 (10% CR): GA	.415	–	–	.433	–	–	–	–	–	–	–	–
		MR	.409	–	–	.430	–	–	–	–	–	–	–	–
		FFNN	.438	–	–	.454	–	–	–	–	–	–	–	–
		PNN	.443	–	–	.459	–	–	–	–	–	–	–	–
		GMM	.590	–	–	.608	–	–	–	–	–	–	–	–
Fuzzy--set-based	Use of Fuzzy logic and Word Net [83]	DUC 2002	.485	.529	.504	–	.237	.260	.247	–	–	–	–	–
	Using Fuzzy Interference [25]	Proceedings of JAIR	.60	.58	.59	–	–	–	–	–	–	–	–	–
Centroid--based	Word embedding with skip gram [68]	DUC 2004	–	–	–	.504	–	–	–	.133	–	–	–	–
Classifi-cation--based	Structural domains, regulated processes [50]	with stopwords	.188	.681	.252	–	–	–	–	–	–	.338	.117	–
		without stopwords	.147	.587	.201	–	–	–	–	–	–	.262	.084	–

8.2. Comparison of abstractive summarization methods

The results of various abstractive summarization methods are listed in Table 2. The semantic representation of text has the greatest score in abstract summarization. It is also seen that graph-based methods and neural network-based methods have comparable rouge scores.

Table 2

Comparison of Abstractive Summarization Methods (Recall and F-Measure of Rouge1, Rouge2, Rouge-SU4, and Rouge-L evaluation metrics)

Type	Method	Dataset	Rouge-1			Rouge-2			Rouge-SU4			Rouge-2	
			P	R	F	P	R	F	P	R	F	R	F
Graph--based	Using concept graph and BabelNet knowledge base [66]	DUC 2004	–	.118	–	–	.084	–	–	.409	–	–	–
	Opinosis [17]	Review from Trip--advisor, Amazon	.448	.283	.327	.142	.085	.099	.226	.085	.103	–	–

Table 2 (cont.)

Type	Method	Dataset	Rouge-1			Rouge-2			Rouge-SU4			Rouge-2	
			P	R	F	P	R	F	P	R	F	R	F
Semantic Representation	Using Semantic Representations [37]	AMR Tree bank	.891	.528	.658	–	–	–	–	–	–	–	–
Neural network--based	Neural Attention Model(ABS+) [69]	DUC 2004	–	.282	–	–	.085	–	–	–	–	.238	–
		Giga word	–	.310	–	–	.127	–	–	–	–	.283	–
	Attentive RNN (RAS-Elman, k=10) [7]	DUC 2004	–	.282	–	–	.085	–	–	–	.238	–	–
	Pointer-generator+ coverage [72]	CNN/ Daily mail	–	–	.395	–	–	–	.173	–	–	.364	–
	Graph-based Attention Neural Model [78]	CNN	–	–	.303	–	–	.098	–	–	–	–	.200
		Daily mail	–	.274	–	–	.113	–	–	–	–	.151	–

In the case of abstractive summary, a low rouge score cannot be directy concluded that the systems generated summary is not that good, as the rouge scores are computed based on the word overlaps between the system's summary and reference summary. An abstractive summary is more likely to have different wordings or word orderings from the original text compared to extractive summary.

8.3. Comparison of journal summarization methods

Comparisons of various journal summarization methods are shown in Tables 3 through 7. In Table 3, the Rouge-L score of different methods is given. Table 4 tabulates the Rouge-2 and Rouge-SU4 scores of statistical methods against abstractive, community, and human summary. Table 5 compares Rouge-1, Rouge-2, and Rouge--SU4 scores of different methods. It is seen that, in journal summarization, the common facts are identified and clustered. The facts can be selected from these different clusters in many ways, like LexRank, C-RR, C-LexRank, Random, etc., to produce a summary covering the maximum number of facts without redundancy. The Pyramid results show that citations can contain useful information that is not available in the abstracts nor the full paper. Similarly abstracts can also contain information not found in citations or full papers. The Pyramid score of summaries from citations and full papers depends on from where the nuggets are chosen for comparison.

Table 3

Results of journal summarization methods in Rouge-L metric

Type	Method	Dataset	Rouge-L					
			smallSAF+F	smallLexRank	smallTextRank	smallKLSum	smallLSA	smallSumBasic
Neural Network--based	Supervised approach to extractive summarization [10]	Science Direct publications	small.32	small.19	small.18	small.17	small.15	small.13

Table 4

Results of Journal summarization methods (Rouge2 and Rouge-SU4 scores against Abstract, Community, and Human summary

Type	Method	Dataset	Rouge-2			Rouge-SU4		
			Abs	Com	Hum	Abs	Com	Hum
Statistical methods	Cross-Document sentence matching, TS (ACL-abs) [2]	CLScisumm	.299	.2	.19	.21	.12	.13

Table 5

Comparison of Journal summarization methods – Precision, Recall, and F-Measure of Rouge1, Rouge2, and Rouge-SU4 evaluation metrics)

Type	Method	Dataset	Rouge-1			Rouge-2			Rouge-SU4		
			P	R	F	P	R	F	P	R	F
Clustering on common facts	Detecting common facts in citation [6]	ANN	.571	.663	.609	.391	.451	.431	–	–	–
	Citation-based summarization using semantic textual similarity [33]	against abstract	–	–	–	–	–	–	.14	.35	.19
		against community summary	–	–	–	–	–	–	.19	.20	.17
Using Context Information	Summarizing citation contexts [52]	DUC 2002: Hierarchical Agglomerative Clustering	–	–	–	–	.150	.389	–	–	–
		Affinity Propagation	–	–	–	–	.151	.268	–	–	–
		LSA cross	–	–	–	–	.111	–	–	–	–
		Hybrid	–	–	–	–	.104	–	–	–	–
	Using citation context and discourse structure (QR-NP: Query reformulation with NP) [8]	QR-NP greedy on CLSciSumm Dataset	–	–	.302	–	–	–	–	–	.257
		QR-NP on TAC Dataset	–	–	.158	–	–	–	–	–	.204
Unsupervised approach	Feature selection, words and minimization model [2]	DUC Aquaint	.131	.415	.189	–	–	–	.09	.56	.144
	Summarization using feature selection approach [2]	DUC2007 Aquaint: without reduction of redundancy	–	–	–	.194	.263	.210	.181	.307	.188
		with reduction of redundancy	–	–	–	.216	.267	.222	.193	.289	.182

Table 6

Results of Journal summarization (Pyramid score of citation summary and context survey)

Type	Method	Dataset	Pyramid	
			Citation summary	Context survey
Using context information	Identifying non explicit citing sentences [64]	AAN QA-CT nuggets	.416	.634
		QA-AB nuggets	.397	.594
		DP-CT nuggets	.324	.379

Table 7

Comparison of Journal summarization (Pyramid score of citation summary and context survey)

Type	Method	Dataset	Pyramid							
			Gold	Random	Lex Rank	C-RR	C-Lex Rank	Trimmer	Mascs	Centroid
Graph and Clustering	Using citation summary networks [63]	AAN	.99	.41	.71	.69	.75	–	–	–
Using citations to generate surveys of scientific paradigms: information content in citation relevance to summary compared to paper and abstract [53]	QA-CT nuggets	Pyramid F Measure on AAN	–	.321	.295	.268	.434	.616	–	–
	QA-AB nuggets	QA citation survey	–	.305	.320	.349	.388	.543	–	–
	QA-CT nuggets	QA abstract survey	–	.452	.441	.480	.383	.404	–	–
	QA-AB nuggets		–	.623	.606	.574	.484	.622	–	–
	QA-CT nuggets	QA full paper survey	–	.239	.190	.299	.446	.199	–	–
	QA-AB nuggets		–	.294	.301	.387	.520	.290	–	–
	DP-CT nuggets	DP citation survey	–	.219	.372	.170	.231	.136	–	–
	DP-CT nuggets	DP abstract survey	–	.321	.311	.263	.301	.312	–	–
	DP-CT nuggets	DP full paper survey	–	.032	*	.144	.000	.280	–	–
Generating Extractive summaries of scientific paradigms [65]	QA-CT nuggets	Pyramid F Measure on AAN	–	.321	.295	.268	.434	–	.616	–
	QA-AB nuggets	QA citations	–	.305	.320	.349	.388	–	.543	–
	QA-CT nuggets	QA abstracts	–	.452	.441	.480	.383	–	.404	–
	QA-AB nuggets		–	.623	.606	.574	.484	–	.622	–
	QA-CT nuggets	QA full papers	–	.239	.190	.299	.446	–	.199	–
	QA-AB nuggets		–	.294	.301	.387	.520	–	.290	–
	DP-CT nuggets	DP citations	–	.219	.372	.170	.231	–	.136	–
	DP-CT nuggets	DP abstracts	–	.321	.311	.263	.301	–	.312	–
	DP-CT nuggets	DP full papers	–	.032	*	.144	.000	–	.280	–
Identifying Keyphrases [28]	Summarizing topics starting from keywords	AAN	–	0.68	0.91	–	0.82	–	–	.610

9. Conclusion

Popular methods of the extractive and abstractive summarization of documents and journal article summarization are discussed in this paper. Fuzzy-based, genetic algorithm-based, and graph-based algorithms are found to have good Rouge scores among the extractive methods. Semantic representation-based and graph-based methods scored higher in the case of abstractive summarization methods.

In the case of journal article summarization, clustering on common facts and using context information gave good Pyramid scores. Extractive method can be used to choose important sentences; then, post-processing techniques can be applied to give a coherent summary. Citation contexts can also be helpful in the clustering of papers and generating the summary. Research activities in journal article summarization are still going on to find an optimal result.

References

[1] Abu-Jbara A., Radev D.: Reference Scope Identification in Citing Sentences. In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 80–90, Association for Computational Linguistics, 2012.

[2] AbuRa'ed A.G.T., Chiruzzo L., Saggion H., Accuosto P., Bravo À.: LaSTUS/TALN @ CLSciSumm-17: Cross-document Sentence Matching and Scientific Text Summarization Systems. In: *BIRNDL@SIGIR*, 2017.

[3] Azam N., Ahmad A.: Text summarization using rough sets. In: *2016 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, pp. 90–94, 2016.

[4] Baralis E., Cagliero L., Mahoto N.A., Fiori A.: GraphSum: Discovering correlations among multiple terms for graph-based summarization, *Information Sciences*, vol. 249, pp. 96–109, 2013.

[5] Barzilay R., Elhadad M.: Using Lexical Chains for Text Summarization. In: *Intelligent Scalable Text Summarization*, 1997.

[6] Chen J., Zhuge H.: Summarization of scientific documents by detecting common facts in citations, *Future Generation Computer Systems*, vol. 32(C), pp. 246–252, 2014.

[7] Chopra S., Auli M., Rush A.M.: Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In: *HLT-NAACL*, 2016.

[8] Cohan A., Goharian N.: Scientific Article Summarization Using Citation-Context and Article's Discourse Structure, *EMNLP*, pp. 390–400, 2015.

[9] Cohan A., Goharian N.: Revisiting Summarization Evaluation for Scientific Articles. In: *ArXiv*, vol. abs/1604.00400, 2016.

[10] Collins E., Augenstein I., Riedel S.: A Supervised Approach to Extractive Summarisation of Scientific Papers. In: *CoNLL*, pp. 195–205, 2017.

[11] Conroy J.M., Davis S.: Vector Space Models for Scientific Document Summarization. In: *VS@HLT-NAACL*, pp. 186–191, 2015.

[12] Conroy J.M., O'Leary D.P.: Text summarization via hidden Markov models. In: *SIGIR '01*, pp. 406–407, 2001.

[13] Deerwester S.C., Dumais S.T., Landauer T.K., Furnas G.W., Harshman R.A.: Indexing by Latent Semantic Analysis, *JASIS*, vol. 41, pp. 391–407, 1990.

[14] Edmundson H.: New Methods in Automatic Extracting, *Journal of ACM*, vol. 16(2), pp. 264–285, 1969.

[15] Erkan G., Radev D.R.: LexRank: Graph-based Lexical Centrality as Salience in Text Summarization, *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.

[16] Fattah M.A., Ren F.: GA, MR, FFNN, PNN and GMM based models for automatic text summarization, *Computer Speech & Language*, vol. 23(1), pp. 126–144, 2009.

[17] Ganesan K., Zhai C., Han J.: Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions. In: *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pp. 340–348, Association for Computational Linguistics, 2010.

[18] Genest P.-E, Lapalme G.: Text Generation for Abstractive Summarization. In: *Proceedings of the Third Text Analysis Conference*, National Institute of Standards and Technology, 2010.

[19] Gong Y., Liu X.: Generic text summarization using relevance measure and latent semantic analysis. In: *SIGIR '01*, pp. 19–25, 2001.

[20] Goyal P., Behera L., Mcginnity T.M.: A Context-Based Word Indexing Model for Document Summarization, *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, pp. 1693–1705, 2013.

[21] Greenbacker C.F.: Towards a Framework for Abstractive Summarization of Multimodal Documents. In: *Proceedings of the ACL 2011 Student Session*, HLT-SS'11, pp. 75–80, Association for Computational Linguistics, 2011.

[22] He Z., Chen C., Bu J., Wang C., Zhang L., Cai D., He X.: Unsupervised document summarization from data reconstruction perspective, *Neurocomputing*, vol. 157, pp. 356–366, 2015.

[23] Hirao T., Nishino M., Yoshida Y., Suzuki J., Yasuda N., Nagata M.: Summarizing a Document by Trimming the Discourse Tree, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23(11), pp. 2081–2092, 2015.

[24] Hovy E.: Text Summarization. In: R. Mitkov (ed.), *The Oxford Handbook of Computational Linguistics*, chap. 32, pp. 583–598, Oxford University Press, 2003.

[25] Jafari M., Wang J., Qin Y., Gheisari M., Shahabi A.S., Tao X.: Automatic text summarization using fuzzy inference. In: *2016 22nd International Conference on Automation and Computing (ICAC)*, pp. 256–260, 2016.

[26] Jaidka K., Chandrasekaran M.K., Rustagi S., Kan M.Y.: Insights from CL-SciSumm 2016: the faceted scientific document summarization shared task, *International Journal on Digital Libraries*, vol. 19(2-3), pp. 163–171, 2018.

[27] Jaidka K., Yasunaga M., Chandrasekaran M.K., Radev D.R., Kan M.Y.: The CL-SciSumm Shared Task 2018: Results and Key Insights. In: *BIRNDL@SIGIR*, 2018.

[28] Jha R., Abu-Jbara A., Radev D.: A System for Summarizing Scientific Topics Starting from Keywords. In: *Proceedings of 51st Annual Meeting of the ACL*, pp. 572–577, 2013.

[29] Kaikhah K.: Automatic text summarization with neural networks, *2004 2nd International IEEE Conference on 'Intelligent Systems'. Proceedings (IEEE Cat. No.04EX791)*, vol. 1, pp. 40–44, 2004.

[30] Kallimani J.S., Srinivasa K.G., Reddy B.E.: Statistical and Analytical Study of Guided Abstractive Text Summarization, *Current Science*, vol. 110(1), 2016.

[31] Khan A., Salim N.: A Review on Abstractive Summarization Methods, *Journal of Theoretical and Applied Information Technology*, vol. 59(1), pp. 64–72, 2014.

[32] Kupiec J., Pedersen J.O., Chen F.: A Trainable Document Summarizer. In: *SIGIR '95*, pp. 68–73, 1995.

[33] Lauscher A., Glavas G., Eckert K.: University of Mannheim @ CLSciSumm-17: Citation-Based Summarization of Scientific Articles Using Semantic Textual Similarity. In: *BIRNDL@SIGIR*, 2017.

[34] Le H.T., Le T.M.: An approach to abstractive text summarization. In: *2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR)*, pp. 371–376, 2013.

[35] Lin C.Y.: Training a selection function for extraction. In: *Proceedings of the Eighth International Conference on Information and Knowledge Management*, CIKM'99, pp. 55–62, Association for Computing Machinery, 1999.

[36] Lin C.Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: *Text Summarization Branches Out*, pp. 74–81, Association for Computational Linguistics, 2004.

[37] Liu F., Flanigan J., Thomson S., Sadeh N.M., Smith N.A.: Toward Abstractive Summarization Using Semantic Representations. In: *HLT-NAACL*, 2015.

[38] Lloret E., Palomar M.: Analyzing the Use of Word Graphs for Abstractive Text Summarization. In: *Advances in Information Mining and Management*, 2011.

[39] Lloret E., Romá-Ferri M.T., Palomar M.: COMPENDIUM: a text summarization system for generating abstracts of research papers, *Data & Knowledge Engineering*, vol. 88, pp. 164–175, 2013.

[40] Louis A., Joshi A.K., Nenkova A.: Discourse indicators for content selection in summarization. In: *SIGDIAL Conference*, pp. 147–156, 2010.

[41] Louis A., Nenkova A.: Automatically Evaluating Content Selection in Summarization without Human Models. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing,* vol. 1, pp. 306–314, Association for Computational Linguistics, 2009.

[42] Luhn H.: The Automatic Creation of Literature Abstracts, *IBM Journal of Research and Development*, vol. 2(2), pp. 159–165, 1958.

[43] Mani I., Bloedorn E., Gates B.: Using cohesion and coherence models for text summarization. In: *AAAI 1998*, pp. 69–76, 1998.

[44] Mani I., House D., Klein G., Hirschman L., Firmin T., Sundheim B.: The TIPSTER SUMMAC Text Summarization Evaluation. In: *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, pp. 77–85, Association for Computational Linguistics, 1999.

[45] Mann W., Thompson S.: Rhetorical Structure Theory: Towards a functional theory of text organization, *Information Processing and Management*, vol. 8(3), pp. 243–281, 1988.

[46] Marcu D.: Improving summarization through rhetorical parsing tuning. In: *Sixth Workshop on Very Large Corpora*, pp. 206–215, 1998.

[47] Marcu D.: *The Theory and Practice of Discourse Parsing and Summarization*, MIT Press, 2000.

[48] Marcu D.C.: *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*, Ph.D. thesis, 1998.

[49] Mei Q., Zhai C.: Generating Impact-Based Summaries for Scientific Literature. In: *ACL*, pp. 816–824, 2008.

[50] Méndez-Cruz C.F., Gama-Castro S., Mejía-Almonte C., Castillo-Villalba M.P., Muñiz-Rascado L., Collado-Vides J.: First steps in automatic summarization of transcription factor properties for RegulonDB: classification of sentences about structural domains and regulated processes. In: *Database*, 2017.

[51] Mitra M., Singhal A., Buckley C.: Automatic Text Summarization by Paragraph Extraction. In: *Intelligent Scalable Text Summarization*, 1997.

[52] Mitrovic S., Müller H.: Summarizing Citation Contexts of Scientific Publications. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction – 6th International Conference of the CLEF Association, CLEF*, pp. 154–165, 2015.

[53] Mohammad S., Dorr B., Egan M., Hassan A., Muthukrishan P., Qazvinian V., Radev D., Zajic D.: Using Citations to Generate Surveys of Scientific Paradigms. In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pp. 584–592, Association for Computational Linguistics, 2009.

[54] Molina A., Torres-Moreno J.M., SanJuan E., da Cunha I., Sierra Martínez G.E.: Discursive Sentence Compression. In: *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing*, vol. 2, CICLing'13, pp. 394–407, Springer-Verlag, 2013.

[55] Moratanch N., Chitrakala S.: A survey on abstractive text summarization. In: *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pp. 1–7, 2016.

[56] Nenkova A., Passonneau R., McKeown K.: The Pyramid Method: Incorporating human content selection variation in summarization evaluation, *ACM Transactions on Speech Language Processing*, vol. 4(2), 2007.

[57] Paice C.: Constructing literature abstracts by computer: Techniques and prospects, *Information Processing and Management*, vol. 26(1), pp. 171–186, 1990.

[58] Pal A.R., Maiti P.K., Saha D.: An Approach to Automatic Text Summarization Using Simplified Lesk Algorithm and Wordnet, *International Journal of Control Theory and Computer Modeling*, vol. 3, pp. 15–23, 2013.

[59] Parveen D., Strube M.: Integrating Importance, Non-Redundancy and Coherence in Graph-Based Extractive Summarization. In: *IJCAI*, 2015.

[60] Patil S.R., Mahajan S.: Optimized Summarization of Research Papers as an Aid for Research Scholars Using Data Mining Techniques. In: *2012 International Conference on Radar, Communication and Computing (ICRCC)*, pp. 243–249, 2012.

[61] Pourvali M., Abadeh M.S.: A new graph based text segmentation using Wikipedia for automatic text summarization, *International Journal of Advanced Computer Science and Applications*, vol. 3(1), pp. 35–39, 2012.

[62] Qazvinian V., Hassanabadi L.S., Halavati R.: Summarizing text with a genetic algorithm-based sentence extraction, *International Journal of Knowledge Managemant Studies*, vol. 2(4), 2008.

[63] Qazvinian V., Radev D.R.: Scientific Paper Summarization Using Citation Summary Networks. In: *Proceedings of the 22nd International Conference on Computational Linguistics*, vol. 1, COLING'08, pp. 689–696, Association for Computational Linguistics, 2008.

[64] Qazvinian V., Radev D.R.: Identifying Non-Explicit Citing Sentences for Citation-Based Summarization. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pp. 555–564. Association for Computational Linguistics, 2010.

[65] Qazvinian V., Radev D.R., Mohammad S.M., Dorr B., Zajic D., Whidby M., Moon T.: Generating Extractive Summaries of Scientific Paradigms, *Journal of Artificial Intelligence Research*, vol. 46(1), pp. 165–201, 2013.

[66] Rashidghalam H., Taherkhani M., Mahmoudi F.: Text summarization using concept graph and BabelNet knowledge base. In: *2016 Artificial Intelligence and Robotics (IRANOPEN)*, pp. 115–119, 2016.

[67] Rasim A., Ramiz A.: Evolutionary Algorithm for Extractive Text Summarization, *Intelligent Information Management*, vol. 1(2), pp. 128–138, 2009.

[68] Rossiello G., Basile P., Semeraro G.: Centroid-based Text Summarization through Compositionality of Word Embeddings. In: *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pp. 12–21, Association for Computational Linguistics, 2017.

[69] Rush A.M., Chopra S., Weston J.: A Neural Attention Model for Abstractive Sentence Summarization. In: *EMNLP*, 2015.

[70] Saggion H., AbuRa'ed A., Ronzano F.: Trainable Citation-enhanced Summarization of Scientific Articles. In: *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pp. 175–186, 2016.

[71] Saied H.A., Dugué N., Lamirel J.C.: Automatic summarization of scientific publications using a feature selection approach, *International Journal on Digital Libraries*, vol. 19, pp. 203–215, 2017.

[72] See A., Liu P.J., Manning C.D.: Get To The Point: Summarization with Pointer-Generator Networks. In: *ACL*, 2017.

[73] Shi L., Tong H., Tang J., Lin C.: VEGAS: Visual influEnce GrAph Summarization on Citation Networks, *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, pp. 3417–3431, 2015.

[74] Shi T., Keneshloo Y., Ramakrishnan N., Reddy C.K.: Neural Abstractive Text Summarization with Sequence-to-Sequence Models. In: *ArXiv*, vol. abs/1812.02303, 2018.

[75] Steinberger J., Jezek K.: Evaluation Measures for Text Summarization. In: *Computing and Informatics*, vol. 28, pp. 251–275, 2009.

[76] Su Y., Sun S., Xuan Y., Shi L.: Influence Visualization of Scientific Paper through Flow-Based Citation Network Summarization. In: *Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, ICDMW '15, pp. 1652–1655, IEEE Computer Society, 2015.

[77] Suanmali L., Binwahlan M.S., Salim N.: Sentence Features Fusion for Text Summarization Using Fuzzy Logic. In: *2009 Ninth International Conference on Hybrid Intelligent Systems*, vol. 1, pp. 142–146, 2009.

[78] Tan J., Wan X., Xiao J.: Abstractive Document Summarization with a Graph--Based Attentional Neural Model. In: *ACL*, pp. 1171–1181, 2017.

[79] Teufel S., Moens M.: Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status, *Computational Linguistics*, vol. 28(4), pp. 409–445, 2002.

[80] Teufel S., Siddharthan A., Tidhar D.: Automatic classification of citation function. In: *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 103–110, Association for Computational Linguistics, 2006.

[81] Vázquez E.V., García-Hernández R.A., Ledeneva Y.: Sentence features relevance for extractive text summarization using genetic algorithms, *Journal of Intelligent and Fuzzy Systems*, vol. 35, pp. 353–365, 2018.

[82] Wang X., Yoshida Y., Hirao T., Sudoh K., Nagata M.: Summarization Based on Task-Oriented Discourse Parsing, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23(8), pp. 1358–1367, 2015.

[83] Yadav J., Meena Y.K.: Use of fuzzy logic and wordnet for improving performance of extractive automatic text summarization. In: *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2071–2077, 2016.

[84] Yan S., Wan X.: SRRRank: leveraging semantic roles for extractive multi-document summarization, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22(12), pp. 2048–2058, 2014.

[85] Zahir al S., Fatima Q., Cenek M.: New Graph-Based Text Summarization Method. In: *2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pp. 396–401, 2015.

[86] Zhang Z., Ge S.S., He H.: Mutual-reinforcement document summarization using embedded graph based sentence clustering for storytelling, *Information Processing and Management*, vol. 48(4), pp. 767–778, 2012.

Affiliations

Sheena Kurian K.
Cochin University of Science and Technology, School of Engineering,
sheenakuriank@gmail.com

Sheena Mathew
Cochin University of Science and Technology, School of Engineering,
sheenamathew@cusat.ac.in

Received: 11.07.2019
Revised: 23.12.2019
Accepted: 23.12.2019