Review on Query focused Multi-Document Summarization (QMDS) with Comparative Analysis

PRASENJEET ROY and SUMAN KUNDU, Department of Computer Science and Engineering, Indian Institute of Technology Jodhpur, India

The problem of query-focused multi-document summarization (QMDS) is to generate a summary from multiple source documents on identical/similar topics based on the query submitted by the users. The paper provided a systematic review of the literature of QMDS. The research works are classified into six major categories based on the summarization methodologies used. Different techniques used for finding query-relevant summaries for different algorithms under each of the six major groups are reported. Further, seventeen evaluation metrics used for evaluating algorithms for text summaries against the human-curated summaries are compiled here in this paper. Extensive experiments are performed on 8 different data sets. Comparative results of 9 methodologies, each representing one of the 6 different groups, are presented. Seven different evaluation metrics are used in the comparative study. It is observed that DL and ML based QMDS methods are performing better in comparison to the other methods.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Natural language generation**; • **Information systems** → **Information retrieval query processing**.

Additional Key Words and Phrases: Query focused multi-document summarization, query relevance

ACM Reference Format:
Prasenjeet Roy and Suman Kundu. 2018. Review on Query focused Multi-Document Summarization (QMDS) with Comparative Analysis. 1, 1 (May 2018), 38 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Text summarization is the process of rewriting a document in brief while maintaining its meaning. When multiple sources of the same topic are used as input, it is called Multi-Document Summarization (MDS). There are two types of summarization, viz., extractive and abstractive. The idea of Abstractive summarization [26, 59, 153, 208, 219] represents the core ideas of the source document using natural language generation, whereas, extractive summarization [81, 122, 173, 225, 226] extracts key sentences out of the source document. Summarization may also be query-specific based on user input, referred to as Query Focused Summarization (QFS). In such cases, the summarizer tries to answer user queries from the document by means of summarization. Similar to text summarization, QFS can also work with multiple documents as input which is called Query-focused Multi-Document Summarization (QMDS) [76, 102, 116, 163, 186, 207]. QMDS aims to answer the user's query taking reference from multiple source documents. It has plethora of applications ranging from intelligent education in schools [214] to search engine technology [190], summarizing scientific documents [182] to trending news articles [179], summarization of viral tweets [120] to biographies [230]. For example, one of the applications of QMDS is to help the students correlate the topics from multiple sources in the innovative learning

Authors' address: Prasenjeet Roy, roy.2@iitj.ac.in; Suman Kundu, suman@iitj.ac.in, Department of Computer Science and Engineering, Indian Institute of Technology Jodhpur, NH 62 Nagaur Road, Karwar, Jodhpur, Rajasthan, India, 342037.

Manuscript submitted to ACM

environment [214]. As multi-document summarization involves many different sources of information, it contains a high level of redundancy. It is challenging to produce the summary in an organized manner maintaining its key aspects from diverse views. An ideal summary needs to have a clear structure, maintaining a gradual transition from the outline of the content to more specific themes. The summary should be coherent, complete, and relevant to the query.

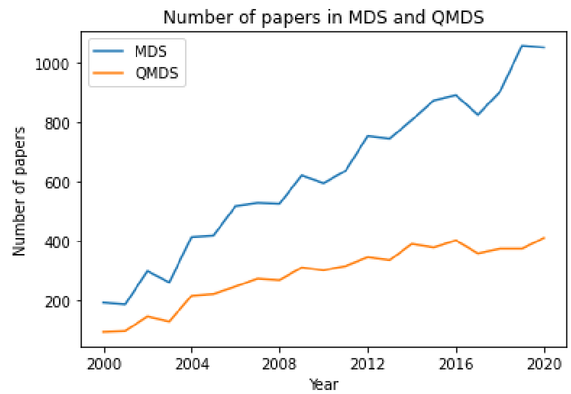


Fig. 1. Plot of number of papers published in MDS and QMDS over the years (Source: Semantic Scholar [53])

Growth in solutions development for MDS and QMDS has been observed in the last two decades. It is evident from the number of publications (Fig. 1). We can infer from the graph that QMDS also has a slightly increasing trend of interest. QMDS was first introduced by Carbonell and Goldstein [23] in 1998. In that seminal work, they introduced Maximal Marginal Relevance (MMR), a ranking method that balances query relevance and novelty of information. That is, it reduces redundancy between the documents. The document would have a higher marginal relevance when relevant to the query and different from already selected documents. MMR is defined as: $\arg\max_{D_i \in R \setminus S} \left[\lambda \left(\mathrm{Sim}_1 \left(D_i, Q\right)\right) - (1-\lambda) \left(\max_{D_j \in S} \mathrm{Sim}_2 \left(D_i, D_j\right)\right)\right]$, where D_i, D_j are the documents, Sim_1 and Sim_2 are similarity measures (could be same or different); Q, λ, R and S represents the query, diversification constant, ranked list of documents and set of selected documents already retrieved in R respectively. Research on QMDS gained its momentum [67, 70, 77, 111, 171, 175, 202, 203, 215] after the release of DUC 2005 dataset on news articles. QMDS is also used for summarizing abstracts of dissertations [88], biographies [230], & question-answers [140] etc..

Text summarization has a rich literature; hence, over the years, there has been a lot of surveys published in this area [42, 43, 55, 60, 66, 78, 143, 183, 192]. Also surveys on XML document summarization [44], documents in Indian regional languages [40, 181], scientific document summarization [96], manifold based techniques [56] and deep-learning-based summarization [127] are available in the literature. There are many surveys on extractive summarization [4, 139], abstractive summarization [85, 138, 159, 164] as well as hybrid summarization [90]. However, all these surveys are on single document text summarization. Similarly, many surveys have been published for MDS [84, 95, 136, 158, 177, 191] in the last decades, including applications specific surveys on MDS [11, 126, 129, 211]. In 2015, Rahman and Borah [162] proposed a survey on QFS; however, no systematic review has been published on QMDS to the best of our knowledge. This motivates us to classify the literature of QMDS with their comparative analysis systematically. In addition to the survey on the methodologies of QMDS, we also enlisted different evaluation metrics used in text summarization problems, including QFS, MDS, QMDS, etc. In the current study, we divided all the methodologies into six groups based on their working principle. The comparative analysis of six methodologies, one from each said group on different datasets, is performed. We used seven different comparative indexes while comparing the performance of these methods. The contributions of this paper can be summarized as:

(1) We classified the literature of QMDS into six groups based on similarities in their text summarization techniques. We also enlisted the query relevancy methodologies used therein.
(2) A detailed comparative study between these groups have been conducted over eight benchmark QMDS datasets using the representative method from each group.
(3) We have compiled seventeen different metrics used for evaluating text summarization. Seven widely used metrics are considered here for the comparative study.

The paper is organized under the following sections. Section 2 reports the recent developments in MDS. Section 3 describes the classification of different QMDS models based on their methodology. Sections 4 and 5 enlists the evaluation metrics and data sets respectively that can be used in the QMDS research. Section 6 reports the comparative study of six groups (using representative methodologies) performed over 8 data sets. Section 7.1 discusses the challenges and future research directions in QMDS. Finally, Section 7 concludes the survey. Table 1 shows the abbreviations used throughout the paper for readers reference.

Table 1. Table of Abbreviations Used

Abbr.	Description	Abbr.	Description
SDS	Single Document Summarization	BIR	Biased Information Richness
MMR	Maximal Marginal Relevance	NER	Named Entity Recognition
BIN	Biased Information Novelty	TF-IDF	Term Frequency- Inverse Document Frequency

2 MULTI-DOCUMENT SUMMARIZATION

There has been massive growth (Fig 1) in the development of solutions for MDS in recent years. Although our primary focus is on QMDS, we discuss briefly the latest developments in MDS considering QMDS is a special case of MDS. Adapting SDS models into MDS carry lots of challenges, such as a larger search space of MDS with limited training data and higher information redundancy in similar documents. In order to solve such issues, 'RL-MMR' [128] uses MMR with guided RL using soft attention for removing redundancy. This way, they generate an extractive summary; however, they lack coherency in the summary. A high-quality summary possesses three essential objectives: importance, redundancy, and length. 'PoBRL' [184] optimize these objectives simultaneously by decoupling them into smaller sub-problems each solved using RL. Similar to RL-MMR, they also used MMR to navigate through the overlapping sentence space of multi-documents. Language-independent statistical learning models are proposed by [13, 87]. The latter also introduced six new features for identifying sentence overlapping and similarity. However, these methods fail to acknowledge the semantic representations of documents. In contrast to this, [199] used a spectral unsupervised MDS where the model uses the affinity matrix generated from document clusters to extract the significant sentences from multi-documents. In the case of a large number of documents, it is computationally heavy to handle the length of the input. It is solved [180] by clustering the documents into disjoint sets and extracting a central representative for each cluster. Li and Zhuge [115] used semantic link networks such as cause-effect and purpose to capture the concepts and events in the input documents. In contrast to this, [3, 222], uses an unsupervised method that converts the documents into a sentence graph and then multi-sentence compression (MSC) [52] to fuse the extractive language units (ELUs) in clusters containing similar core and peripheral articles.

Many researches showed that pre-trained models fine-tuned for SDS can also be used for MDS, e.g., ensemble of single document encoder-decoder is used in [74, 79] to predict the word probabilities based on each document for MDS problem. On the contrary, ‘PRIMER’ [205], an extension of SDS model ‘PEGASUS’ [219] merge multiple documents into single document and use LED model [9] for training. They proposed ‘Entity Pyramid Masking’ for task-oriented pre-training with the Gap Sentence Generation objective. In contrast, [45] proposed a self-supervised method where it trained a supervised model by selecting one of the review documents as the target summary and the remaining ones as the input. In self-supervised settings, hallucinations are more likely due to noise in the training instances. In order to solve this issue, they came up with control tokens that represents the sentiment scores and entities. Similarly, to generate more factual narrative summaries in medical RCTs, [197] train a pipeline model identifying the ‘punchline’ sentences in the input documents. The majority of previous works focused on improving the document representation in the encoder module. In contrast, [155] focused on the decoder module and proposed an attention mechanism based on Determinantal Point Processes [17]. The model can be integrated with any sequence-to-sequence models, from RNNs to transformers, to tackle noisy and longer documents.

We discussed how graphs and encoder-decoder models solve the MDS separately. Recent researches shows that the combination of graphs and pre-trained encoder-decoder models are not only scalable to longer input documents but also process auxiliary additional graphical representations derived from multi-document clusters. Li et al. [113] proposed the first abstractive MDS model that leverages explicit graph representation to process the multi-documents. They incorporated a hierarchical graph-informed attention mechanism to capture cross-document relations in the encoding stage. In [152], dual-encoder i.e., a combination of text encoder and graph encoder is used with pre-trained BART [108]. Frequently appearing entities and their mentions can be significant in making the summary concise and coherent. ‘EMSum’ [229], an entity aware summarization model, augments transformer models with heterogeneous graph for capturing the cross-document information. They incorporated graph attention networks to capture the flow of information between the nodes. Simply concatenating multi-documents into a flat sequence loses the hierarchical structure of the document clusters. Hence, [80] treats documents, sentences, and words at three granular levels in a hierarchical multi-granularity interaction network. The proposed model could produce both extractive and abstractive summarization. However, this may lead to a loss of fine-grained interaction between the features. On the other hand, multi-documents are viewed as heterogeneous graphs at different granularities in [34]. It uses a graph-to-sequence framework for generating summaries. In order to distill salient information from multi-documents, they jointly optimize a neural topic model (NTM) and an abstractive summarizer to incorporate latent topics in the summary generation. In contrast, ‘SgSum’ [25] models MDS as a sub-graph selection problem; the input is in the form of relation graphs and their candidate summaries as sub-graphs.

Task-oriented pre-training helps in refining the pre-train setup that closely resembles the downstream task. In [201], a combination of task-agnostic pre-trained language models and task-specific priors improved the performance in low-resource settings. This boosts the performance by filtering out task-irrelevant patterns and enhancing task-specific information during fine-tuning. As opposed to this, [141] incorporated entity-level content planning as a pre-training objective into PEGASUS for summary generation and content-level planning. They created an augmented target summary by prepending the entity chain in summary that could control hallucinations in an abstractive summary. In [232], three pre-training objectives sentence reordering, next sentence generation, and masked document generation are used to train sequence-to-sequence models for abstractive summarization on unlabeled texts. Although task-specific pre-training help in sentence selection in extractive datasets, it does not reflect much improvement on abstractive datasets [172].

3 CLASSIFICATION OF QMDS METHODOLOGIES

We have classified the QMDS methodologies into six different primary groups and a few subgroups as shown in Fig. 2 based on their approaches. In this section, we describe these groups in detail.

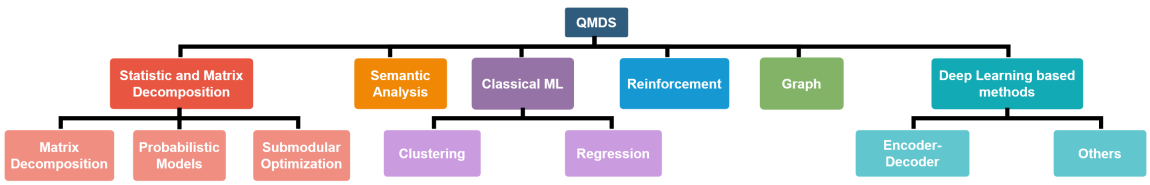


Fig. 2. Methodology based classification of QMDS models

3.1 Semantic analysis based methods

The semantic analysis defines how different syntactic structures such as phrases, sentences, or documents are interlinked to form independent language meanings. This way, it solves one of the critical challenges of QMDS by identifying semantic relatedness between the sentences and given query. The development of different methodologies over time in this group is shown in Fig. 3.

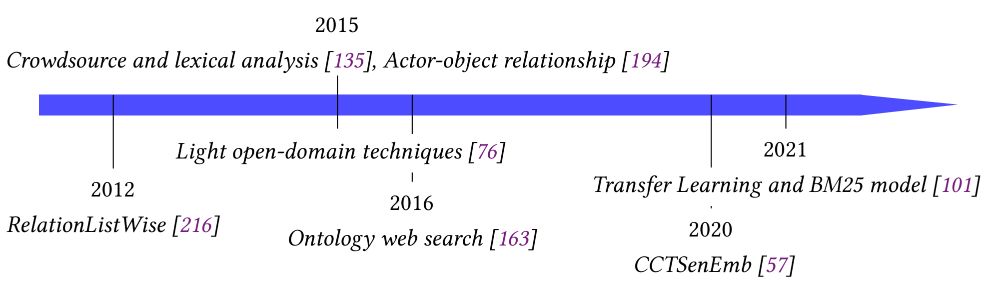


Fig. 3. Timeline of semantic analysis based methods

'RelationListwise' [216] captures the relation between sentences using maximizing estimated likelihood. They used a log-bilinear probabilistic distribution to capture the semantic relatedness between the terms. The authors constructed a word connectivity graph along with the PageRank [206] algorithm to measure the word importance. The authors have incorporated query relevance using BIR and BIN. Here, BIR quantifies query-related information contained in the sentence using manifold ranking, whereas BIN focuses on information redundancy while capturing the information needed for the query using DivRank. The results ignore the sentence-to-sentence similarity while incorporating the summary generation. Unlike previous work, [135] determine the query-to-sentence and sentence-to-sentence similarities using crowdsourcing and lexical-semantic resources. Authors extended the traditional WordNet-based similarity [132] approach by converting the POS tags such as verbs, adjectives, and adverbs into equivalent nouns using CatVar and Morphosemantic links [51]. However, this simple lookup conversion is challenging with WordNet lexical organization. To compute query-sentence similarity, they integrated Wikipedia, Wordnet, and NER query relevances along with two scoring parameters a) Subsumed Semantic Content (SSC) and b) Centroid. MMR is used to generate the final summary to avoid redundancy. These models use complex methods for compression and extraction of sentences and hence, take more computational time.

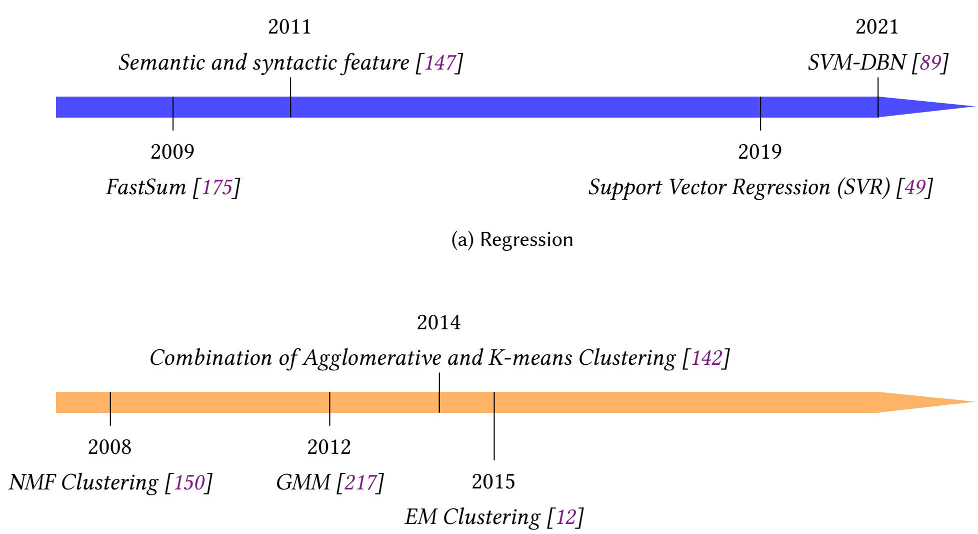
In order to solve this issue, [76] focused on semantic literature and light-weight open domain techniques. They proposed two approaches for handling multiple documents, first, by aggregating SDS using linear semantic analysis to do QMDS, and second, by semantic triples clustering with focusing overlap between the n-grams. The most focused salient triple for summary is obtained by performing semantic overlap of sentences with the query. The semantic triples capture the mutual meaning between the sentences making the model easier to extract the focused sentences. These focused sentences are scored based on their overlap with the given query. Inspired by sentence-to-sentence features, [194] uses an ensemble of models to generate a ranking of sentences. According to them, sentences showing actor-object relationships can better correlate with the query. Hence, Stanford parser is used to give more weightage to those sentences that consist of subject and object clauses. The authors have used query-dependent features such as word, semantic, and named-entity similarities to incorporate sentence relevance to the query. On the other hand, 'NUCLEUS' [163] uses ontology and Web Search Query Log (WSQL) to identify the most frequent queries for each group. WSQL helps identify the user's search preferences to address the query better. Ontology helps in recognizing the salient entities or keywords in the sentences. NUCLEUS also generates new query terms using ontology to enrich meaning in the generated summary. The previous representations of sentences do not maintain order and semantic relationships between the words in a sentence which also carry the meaning. So, [101] utilized the pre-trained embedding models to capture the syntactic and semantic relationships between the words. They combined BM25 [169] and semantic similarity functions to compute query relevance score. Sentence saliency is essential in identifying the necessary aspects of a document. CCTSenEmb [57] used discriminative topics to incorporate sentence and topic embeddings to predict subsequent sentence representation.

3.2 Classical ML based methods

Semantic analysis-based methods use sentence-level and phrase-level features separately for the rank of the sentences. On the other hand, the ML-based methods try to learn the best combination of different sentence features to better rank sentences. We have divided Classical ML methods into two subgroups, i.e., regression and clustering. The timeline of these methods is shown in Fig. 4.

3.2.1 Regression based methods. In contrast to semantic analysis-based methods, regression models use training samples to learn continuous functions for a good approximation of relevance of the sentences concerning the query. 'Fastsum' [175] uses the word-level and sentence-level features to decide the topic description in summary. They used topic-title and topic-description as sentence features for incorporating query relevances. The main observation of the work is that document frequency, and topic-title frequency is essential features for ranking sentences. However, the limitation is that it is based only on term-frequency features and does not consider the semantic analysis of words. This problem was solved in [147] and [49] by encompassing semantic features along with syntactic features for better extraction of sentences from the respective documents. In addition to previous sentence-level features, the features consisted of query-focused semantic matching feature, NER feature, TF-IDF, and stop-word penalty feature. This way, they incorporated query relevance with the sentences. However, if the threshold at the final summary size is too small, it lacks correlation between the summarized sentences. The extractive method chooses the high-ranking sentences while losing the topic's essence in the average ranked sentences. In contrast to this, 'SVM-DBN' [89] is a hybridization of deep belief networks (DBN) with SVM. Their feature space includes TF, sentence-topic similarity, temporal difference (td) and sentence penalty. DBM helps in fine-tuning the resulting classification from SVM.

3.2.2 Clustering based methods. Researchers used clustering methods to avoid redundancy and biasing in the inherent semantics in the documents. Cluster-level information helps in the ranking of sentences for the final summary. According to Park et al. [150], humans use only non-negative part of the information in their cognitive mind, and they proposed a clustering-based method considering the same. The authors extracted the semantic features from the sentences using Non-negative Matrix Factorization (NMF) clustering. In this way, the original TF-IDF matrix is decomposed into the semantic feature and the semantic variable matrix. It is the property of NMF to determine the inherent structure of the documents. The semantically related terms are grouped into semantic features, followed by ranking and summary generation. The semantic feature matrix captures the most significant cosine similarity value concerning the query. On the other hand, [217] make it more semantically relevant by ranking the sentences in four relevant features viz title, document, query, and cluster. These are represented in the latent topic vector space model [174]. The relevance similarity is computed using JS divergence [100]. The BIR mechanism calculates the relevance score, and the redundancy is avoided using BIN. Gaussian Mixture Model [167] is used to help in regulating the size of the target cluster to make the ranking of sentences more robust by ignoring the outliers. The results shown in the original paper explain that sentence-query relevance with BIN had a significant effect on the quality of the summary, but it is computationally expensive if the data is too large. In [142], a hybrid method combining the agglomerative [204] and K-means clustering [54] is used to capture the topic groups matching with the query. Bhagat and Ingle [12] used the expectation-maximization approach to observe the less observed terms in sentences because to make the extractive summarization more coherent, the terms less used are also essential on making the final summary meaningful. The query relevance is incorporated by mutually



(a) Regression

(b) Clustering

Fig. 4. Timeline of classical ML based methods

reinforcing query and the sentence clusters. The higher-ranked sentences are selected for candidate summary after the convergence.

3.3 Statistic and matrix decomposition (SMD) based methods

The classical ML models do not exploit the intrinsic structure of the sentences. We have not explored how intrinsic topics or themes in the query can identify the candidate sentences. Statistical models are used to identify clear interpretations about the themes and corresponding sentences similar to the given query. This group consists of methods related to matrix decomposition, probabilistic models, submodular optimization, and the timelines are shown in Fig. 5.

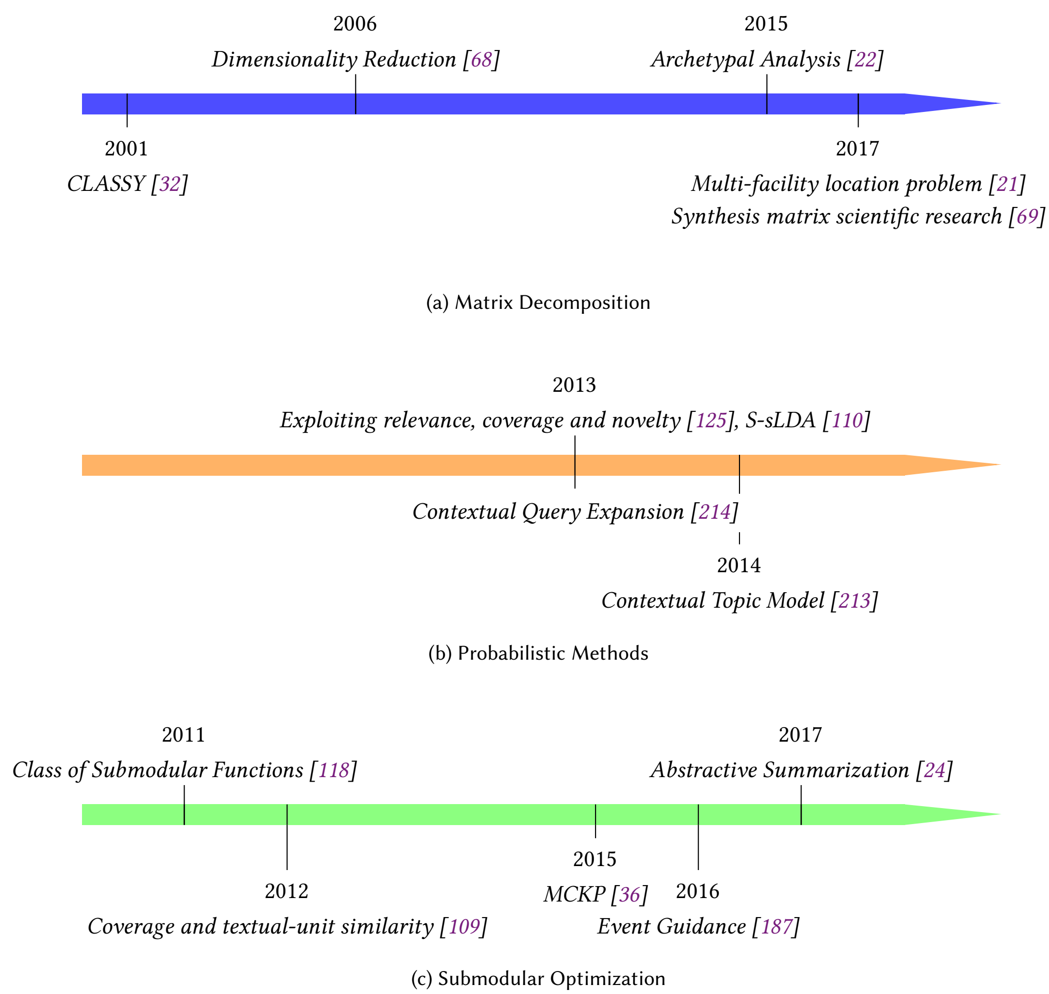


Fig. 5. Timeline of statistic and matrix decomposition based methods

3.3.1 Matrix Decomposition based methods. When we model the documents using the semantic features of the sentences, we get a large distributed semantic space matrix. It is quite challenging to work with these high-dimensional matrices, and matrix decomposition is one of the solutions for the same. CLASSY [32] was one of the pioneer's works in QMDS using matrix decomposition. In this work, the query terms are selected based on their POS tags and named entities. It uses Hidden Markov Model (HMM) [6] to score the sentences and then pivoted QR decomposition [64] to produce the minimum redundant sentences as output. However, the quality of generated summary depends upon the identified named entities. In contrast to CLASSY, [68] uses Singular Value Decomposition [63] for decomposing co-occurrence term matrix. The query relevance is incorporated by computing cosine similarity between the sentences and query in the sentence extraction algorithm. Then, the MMR is used to select the candidate sentences for the final summary avoiding redundancy. Canhasi and Kononenko [22], on the other hand, used a combination of convex NMF and weighted Archetypal Analysis [35] to cluster and rank the relevant sentences from the similarity matrix. They designed a similarity graph with a weighted matrix to incorporate the relevance of the query in the document. 'mFLSum' [21] further uses linear programming [233] to address the problem as a multi-facility problem. These algorithms do not use term-level relations like n-grams or phrases in their pre-processing phases. In [69], a synthesis-based approach is used to perform summarization in two steps, first sentence selection done by aspect analysis of each sentence, and second, ranking using query-focused LexRank (Q-LexRank). Q-LexRank is a modified version of LexRank [46] which consists of query relevance scores as edge weights to give importance to sentences that are more correlated to the query. All of the above matrix decomposition-based methods output an extractive summary of the user query. As per our best knowledge, no attempt was made for abstractive summarization of QMDS using matrix decomposition.

3.3.2 Probabilistic methods. Bayesian models give clear probabilistic interpretations exploiting the intrinsic structure of the sentences for the summary generations. One of the popular bayesian models is Latent Dirichlet Allocation (LDA) [16] that uses latent topics to describe the observations. 'S-sLDA' [110] is a sentence-feature based supervised LDA [15] for solving QMDS. It combines supervised approaches and topic modeling to learn optimum feature weights. They assumed that words in the same sentence belonged to the same topic. The generative process of S-sLDA uses word features from the current sentences as well as neighboring sentences. They design a learning strategy that computes the probability of all tokens concerning the query generated from the corpus. In the training phase, the human assessors label the sentences with scores. The feature space consists of cosine similarity with the query, local Inner-Document Degree order (IDD), and other typical sentence and document features. The labeled set helps learn the weights of these features for the summary generation of target datasets. The sentences extracted are shorter in length with correct information due to topic modeling of feature space. Similar to this, [214] combine topical n-gram model used in information retrieval and query likelihood (QL) model to generate the contextual topics from the bigram distribution. The former identifies key phrases from contexts, and the latter recognizes semantic correlations between them to extract meaningful sentences relevant to the query. They utilized the Expected Mutual Information Measure (EMIM) [33] to choose the correct topic words for the context. The updated contextual topics are passed to a QL-based ranking algorithm for scoring the sentences and integrated with MMR to generate the final summary. The generated summary is coherent due to the meaningful phrases extracted during conceptual modeling. The sentence selection strategy improved further by hierarchical topic model and deep statistical analysis [213].

An ideal summary is supposed to maintain a reasonable balance between novelty, coverage, and relevance to the query. 'PRCN' [125] covers these features using a mixture of Probabilistic Latent Semantic Analysis (PLSA) [73] and Probabilistic Hyperlink-Induced Search (PHITS) [30]. PLSA provides a probabilistic understanding of word co-occurrence based on latent topic space, whereas PHITS make inter-sentence links exploiting sentence similarity for model generation. The framework is a joint probabilistic model covering relevance and coverage on topics. They achieved topic relevance by computing cosine similarities between the sentences and topic coverage using the term-frequency matrix. These two models combine to form a reference topic model. The feature space consists of document novelty, query novelty, document perspective, and query perspective. They further proposed a greedy algorithm for generating a summary balancing the topics and query. They found that document novelty and query novelty are the most essential features.

3.3.3 Submodular optimization methods. One of the critical challenges of QMDS is to integrate query relevance and coverage while avoiding redundancy. Lin and Bilmes [118] proposed one of the pioneering works in QMDS using submodular optimization, where query relevance in summary is incorporated by a class of submodular functions. It maintains a trade-off between coverage and diversity. The objective function is modeled as a knapsack constraint problem. Coverage measures how similar the summarised set and the original document is, whereas diversity rewards the sentence estimating its importance in summary. Both these functions preserve monotonicity and submodularity. In contrast to the previous work where authors have used only sentence similarity, [109] also considered term coverage to extract more granularity in the context. They have modeled the summarization problem as a budgeted maximum coverage problem. The authors have designed a greedy algorithm to take advantage of submodularity. Together with term coverage and textual similarity, they also used MMR to reduce redundancy and maintain query relevance in the generated summary. To further improve its running time, [109] use the accelerated greedy algorithm [133]. Davoodi and Chali [36] introduced compression with a semi-extractive maximum knapsack coverage problem which was lacking in the previous literature. Unlike previous methods where word-matching is used for query relevance, here, authors have employed WordNet-based semantic similarity measures to employ query relevance. They used Berkeley parser to generate the parse tree for each sentence, and then for compression, they used Berg's compression [10] method to detect the deletable terms in the sentence. The objective function consists of maximizing the three measures viz coverage, relevance, and compression.

All the aforementioned works are extractive summarization models, which are pretty different from human-annotated summaries. To solve this issue, [187] proposed an abstractive method that is divided into two parts, first, sentence clustering, and second, multi-sentence compression algorithm. The events are extracted from the sentences in the form of a tuple (Subject, Predicate, Object) using Stanford Parser [92]. For example, "The college delayed the upcoming exams." is (college, delayed, exams). These are embedded as a distributed feature vector as $\overrightarrow{Sub, Verb, Obj} = \overrightarrow{Verb} \odot (\overrightarrow{Sub} \otimes \overrightarrow{Obj})$. The Chinese Whispers method [14] is used for clustering as it is a randomized graph-based algorithm with high scalability. After the clustering, the candidate sentences are generated using a word graph. The word pairs are generated for all pairs of sentences; the most common vertices increase the fusion probability for condensed sentence representation. The vertices are chosen based on their distance to the centroid event. Similar to [118], [187] have incorporated query relevance in the objective function in addition to topic coverage and diversity between the sentences.

Similar to [36], [24] also used the compression function with an addition of merging function. The compression method helped remove the nominal terms and later applied the merging function to join two sentences beginning with a common coreferent subject. Stanford Coreference Resolution engine [161] is being used to generate noun phrases in the document. Sentences having similar coreferent noun-phrase but dissimilar verb-phrase are merged. In addition to query relevance in the objective function, importance and non-redundancy metrics are also incorporated.

3.4 Reinforcement based methods

In QMDS, our task is to train the model to extract meaningful sentences from multiple documents relevant to the given query. So, it is desirable to increase this likelihood of extracting only semantically correlated sentences from the documents. This can be achieved by giving rewards in a reinforcement manner. Timeline of reinforcement based methods for QMDS is shown in Fig. 6.

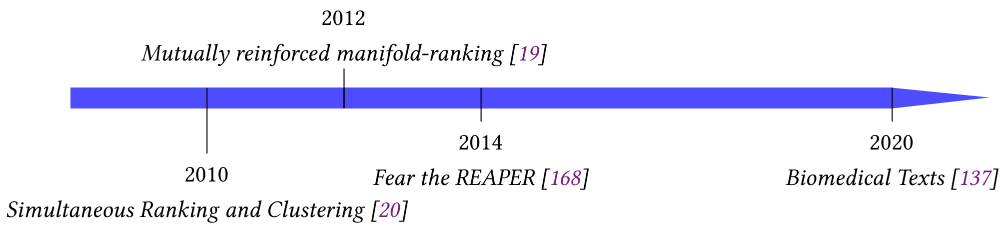


Fig. 6. Timeline of reinforcement based methods

In statistic-based methods, we have observed that sentences are ranked and clustered independently, lacking coordination between them. Cai et al. [20] proposed a novel approach to simultaneously rank sentences and clusters using RL. RL explores the rank distribution of sentences and terms over the discovered clusters. The authors developed a document bi-type graph between sentences and the terms associated with the sentences. Three ranking functions are proposed viz (a) global rank, which relies only on sentence ranking, (b) local-rank ranks the sentences within clusters, and (c) conditional rank computes the rank distribution of sentences and terms in cluster. The ranking function calculates the ensembled conditional ranking scores of all the sentences. The query relevance is imposed using cosine similarity between the theme cluster and the query tokens. This model lacks semantic relationships between the terms.

While the majority of the aforementioned works calculate relevance based on sentences only, they avoid the document-level information that helps in understanding the content and how it influences the ranking. However, the rank of a sentence depends on relevance with the query as well as relevance between the query and document [19]. Hence, sentences of documents having higher relevance to the query are ranked higher and the rank of a document is high if it contains sentences that have more relevance to the query. They proposed a two-layer graph linking both sentences and documents and using it in the proposed 'Mutual Reinforced Relevant Propagation' (MR2P). This architecture would help in focusing the coverage content of the source documents. The authors have explored the relationship between sentence-to-sentence and document-to-document, while the sentence-to-document relationship have not been explored.

'REAPER' [168] primarily used TD(λ), SARSA, and Approximate Policy iteration-based methods for exploration. The feature set comprised of coverage ratio, redundancy ratio, length ratio, longest common subsequence (LCS), etc. The reward function of REAPER is based on the concurrence score and LCS recall metric. They proposed query-focused rewards to give preference to the sentences related to the query while maintaining the trade-off between overall similarity and query similarity score. The model worked well for MDS and could improve QMDS by applying some disambiguation methods while ranking the sentences relevant to the query. In 2020, [137] utilized RL methods in biomedical texts. Instead of policy iterations used in the previous work, they have used Proximal Policy Optimisation (PPO) [176] approach for summarization. They incorporated five components such as candidate sentences, questions, summary generated so far, sentences after respective candidate sentence, and entire document together as input in the neural architecture of PPO.

3.5 Graph based methods

In text analysis of web information, graphs are widely used to find insightful information from complex structures. Later, it is also adapted in the field of text summarization to identify which edges are highly correlated to the given query [82, 83, 86, 99, 165]. The development of QMDS graph-based methods over the years is shown in Fig. 7.

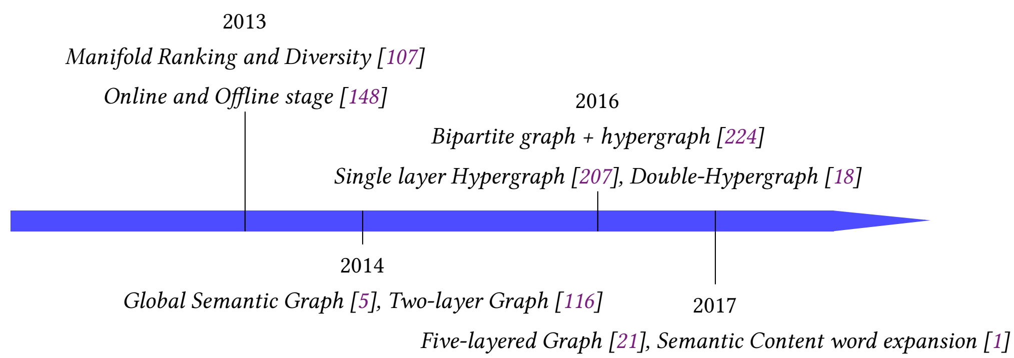


Fig. 7. Timeline of graph based methods

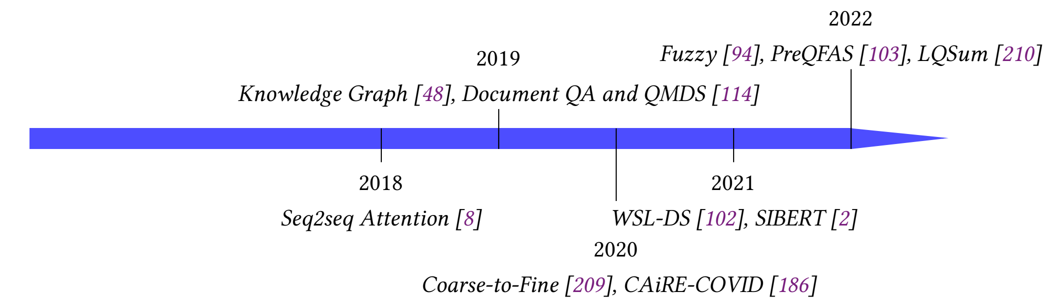
Pandit and Potey [148] used a graph-based framework that has two stages, offline and online. The offline stage considers paragraphs as nodes of the document graph, and the online stage gives query-specific weights to each node. They designed a weighted clustered document graph with edge weights as TF-IDF scores to get a query-focused summary. A minimum spanning tree is used for a keyword search to get the relevant path as a summary to the query. In contrast to this, [107] have used the manifold ranking method along with DivRank [130] to focus both on relevance propagation as well as diversity in summarization. The query node is initialized with 1 and other sentences as 0; this way, they spread their influence to their neighboring nodes and we extract the query-aware sentences at convergence. The algorithm follows rich get richer phenomena as the nodes which are visited maximum times during random walk tend to have a higher weight at the following walk. On the other hand, [5] focused on two targets, first, achieving non-redundancy by graph matching of semantic and syntactic features of the semantic graph, and second, query relevance by integrating concept similarities with a modified spreading algorithm to choose the shortest path. A two-layer, namely, a topic and a sentence layer graph structure, is used in [116]. A query is also included in the sentence layer as a node. LDA topic modeling is performed at word-level as well as sentence-level. As background and document-specific information influence the quality of topic modeling, it performs well for capturing the semantic similarity with the query terms. They iteratively rank the sentences with respect to the query node to incorporate query relevance. The prediction of the optimal number of topics in LDA was a challenge as it changed how the final summary is generated.

The aforementioned works focused mainly on the syntactic features instead of semantics. ‘QSLK’ [1] solves QMDS using linguistic knowledge database and word semantics. The model creates a word set for computing semantic vectors and word vectors for the sentences. They used WordNet similarity to score the sentences relevant to the given query. It incorporates the Content Word Expansion method for expanding the terms and capturing semantic and word order similarity between the sentences and query. However, it cannot distinguish between active and passive sentences as WordNet has limited word coverage for semantic similarity matching. Instead of a simple graph edge, a hypergraph edge can join multiple vertices. A hypergraph framework is used in [207] to capture the word-topic and word-pair similarities within the sentences which reduces the limitations of traversal of random walk in simple graphs. They constructed a query-focused sentence ranking algorithm that takes query-similarity as the reinforced-vertex for random walking. Their topic distribution relationship is restricted to sentences only, and pairwise relationships among documents are not explored. A combination of bipartite graph and hypergraph is used in [224] to extract query-sensitive information. They map the concepts of sentences with the query in a bipartite graph to extract the ranked weights that are used to rank sentences belonging to those concepts in a hypergraph model. In contrast, [18] used a double hypergraph exploiting the sentence-topic and document-topic relationship. They performed Affinity Propagation to cluster the sentences and documents specific to the query using their cosine similarities. A human-annotated summary consists of five essential properties viz purpose of the protagonist, temporal, spatial behavior, cause, and intention of action [234]. Canhasi [21] address these properties using a five-layer graph representing inter and intra relationships among frame layer, sentence layer, query layer, document layer, and paragraph layer. Using a separate query layer linked with the frame and paragraph layer makes the ranking mechanism query-specific to the document.

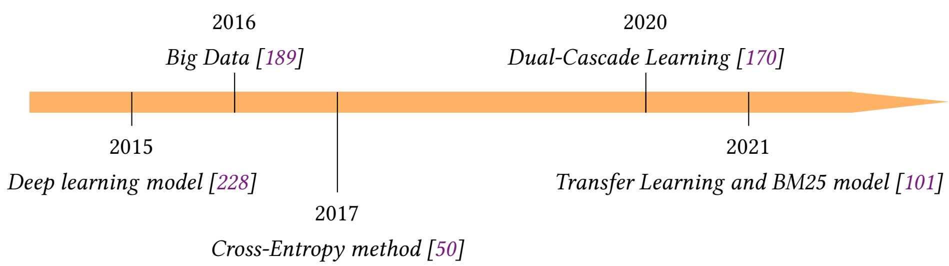
3.6 Deep Learning based methods

Deep learning (DL) models have shown great results in various NLP applications. Due to their scalability, it can generate millions of features for effective representation for encodings. We have divided the group further into two subsections viz encoder-decoder and other deep learning-based methods. Timeline of these subgroups are shown in Fig. 8.

3.6.1 Encoder-decoder model based methods. The previously studied conventional approaches use manual sentence features to extract the relevant sentences, but manual annotation still has certain limitations. However, DL models drastically reduce these dependencies by capturing non-linearities in the data. The publicly available SDS datasets viz CNN/Daily News, Debatepedia is not enough to train the neural network for the QMDS task. Baumel et al. [8] proposed the first DL method that generates abstractive summary in QMDS. It uses a sequence to sequence approach, which has an encoder-decoder model consisting of LSTM [72] with an attention layer to get maximum coverage of the previous information. The query relevance is imposed in two steps; the former is a relevance model that determines the information content relating to the query in the source and later combines these in the coherent summary. They computed Relevance Sensitive Attention (RSA-QFS) using TF-IDF and Word2Vec embeddings for measuring query relevance. The drawback is that it leads to content redundancy and needs improvement in the proper formatting of output sentences. In contrast to this, [48] use a local knowledge graph for each query which compresses the web information avoiding redundancy. These local graphs are later linearized into sequences. In addition to token and position embeddings, they employed graph weight and query relevance embeddings for each generated sequence. To avoid the expensive computations of transformer architecture, they used Memory Compressed Attention (MCA) [119]



(a) Encoder-Decoder Model



(b) Others

Fig. 8. Timeline of deep learning based methods

mechanism in the encoder model. Since the aforementioned datasets are too small to train the neural network, [114] used document-based question answering (DQA) datasets for QMDS models. QMDS can be interpreted as an extension of DQA. They designed a hierarchical encoder-decoder model using a word-level encoder and a document-level encoder. The first one learns representations between the query and the sentences in the document while the latter learns a representation of the document using the BiGRU [28]. They incorporated query relevance using the pre-trained DQA model that provides better semantic information between queries and sentences. This improves the query matching capability while finetuning in the QMDS model. Despite this, it becomes difficult if the document contains a vast number of sentences.

Although using DQA questions for QMDS helps train the models, DQA questions are short and fact-based, whereas QMDS narratives are mostly complex and long; hence, we need to incorporate special attention on queries while retrieval of sentences for the summary. Xu and Lapata [209] focused on the coarse-to-fine technique of estimating the text segments relevant to the query with proper evidence. The model consists of three modules, the relevance estimator to retrieve the text segments relevant to the query, the evidence estimator to measure the semantic similarity between the selected text and query, and finally, a centrality estimator to rank the sentences for a summary generation. The relevance estimator and evidence estimator handle the query relevance. The evidence estimator performs sentence selection and span extraction using BERT [39] to identify the particular span of words in a sentence correctly. In the centrality estimator, an extension of the LexRank algorithm is used to identify the central node to be included in the final summary. This method works well for tasks where a descriptive summary is needed, but it fails to produce a short meaningful context summary. A weakly supervised learning is used in [102] to make the query-focused model trained on the different datasets and use the masking to fine-tune the model on desired domain dataset. The generation of weak reference summary is performed in two steps viz a) Finetune the RoBERTa [123] model in MS-MARCO [145] dataset for answer selection task generating weak extractive summary and then, b) Finetune RoBERTa model in MRPC [41] dataset for paraphrase identification task to measure similarity between weak reference summary and multi-document abstractive gold summaries. This way, it selects the sentences for reference weak abstractive summary. They used an iterative approach of fine-tuning the weak abstractive summary with incorporated queries using the BERTSUM [121] model. The generated output sentences are further fine-tuned on the RoBERTa model in MS-MARCO for selecting the best-ranked sentences. The idea fails to give results if the domain adaptation between the two datasets is from a different distribution. Further, '$PreQFAS_{SFT}$' [103] performs sequential fine-tuning with sentence filtering in the early stage. In the first phase, they identify those sentences that are most relevant to the query in the document set and add them based on their relevance ranking with the query until the allowable token length. In order to incorporate query relevance, this filtered document and query are passed to the BERTSUM model pre-trained on generic abstractive summarization. With a sequential fine-tuning approach, it produces a query-focused abstractive summary. Su et al. [186] proposed a system for question-answering related to COVID-pandemic using the two question answering models viz HLTC-MRQA [185] and BioBERT [106]. They fine-tuned BART on CNN/DailyMail dataset and filtered top-k paragraphs as input to the MDS ranked according to their query-relevance. In order to incorporate query-relevance, they concatenate query at the end of each source paragraph and its respective answer span as input to the BART model. 'SIBERT' [94] produces extractive query focused summaries based on the hierarchical nature of the multi-documents, whereas, [2] uses fuzzy rules with linguistic heuristics to solve QMDS. 'LQSum' [210] uses a generative model for QMDS where it optimizes latent query model and conditional language model.

3.6.2 Other deep learning based methods. Due to the scarcity of labeled training data, the applicability of supervised methods is still a challenging issue. To this end, the solution for QMDS is shifted towards the unsupervised methods. 'QODE' [228] is one of the pioneer works that used unsupervised DL for QMDS. The framework constitutes of three phases viz concept extraction, reconstruction validation, and summary generation. They have utilized RBMs with Gibbs sampling for building each layer block. In concept extraction phase, they have three hidden layers to filter out irrelevant words, identify keywords, and extract candidate sentences. The authors utilized the input query in two ways, first, by initializing the weight settings and imposing a penalty in the reconstruction error concerning the query to incorporate query relevance. In second phase, they used back-propagation to fine-tune all the parameters for optimal reconstruction. and generate an importance matrix to calculate the importance score of every sentence. In the final phase, they utilized dynamic programming to obtain the generated summary within the length constraints. A hybridization of feature-based algorithms and dynamic programming is used in [189]. It is used in real-time systems in web searches using Hadoop. The user would input a query, and using Google API, it would fetch top-k URLs. These URLs are taken as input documents and later execute a hybrid feature-based algorithm to generate a summary in the backend. Instead of the MMR, here, dynamic programming is being used to avoid redundancy. To further increase the efficiency, they utilized MapReduce algorithms to handle big data. The Hadoop environment reduced the inference time in a more significant number of documents but performed worse when the number of documents was small.

On the other hand, [50] solves QMDS as a sentence subset selection problem using the cross-entropy (CE) method. CE defines an optimal selection policy to choose the input sentences for the candidate summary. They sampled a series of sentence subsets which chooses the sentences independently for the summary with an initial probability. This probability is updated iteratively, converging to a globally optimal solution. To incorporate query relevance, the authors used six features such as Bhattacharyya similarity between the query and candidate summary set, relative mass devoted by the summary to the query and other sentence features. They further improved its efficiency by pruning the sentences which have higher similarity to the topic description. 'Dual-CES' [170] maintain a trade-off between saliency and focus in the generated summary. It is a two-step optimization approach with distillation to generate saliency-based pseudo feedback. The authors observed that with an increase in summary length, saliency increases while focus decreases. It is an extension of the previous cross-entropy-based approach. Instead of addressing saliency and focus together, it is addressed sequentially in two separate invocations in Dual-CES. Unlike others, they aim to improve the saliency of focused summaries taking distill hints from the human-generated summaries. The first phase of Dual-CES is similar to CES [50] producing a long salient pseudo reference summary. They utilized previously derived predictors in CES such as coverage, position bias, summary length, focus-drift, and an additional predictor asymmetric coverage for higher saliency in the first phase. In the second phase, they have the same input documents with pseudo-reference summary to produce a focused summary keeping the saliency high. In addition to the previous five, they proposed two new predictors, query-relevancy and reference summary coverage, for measuring relevance to the query keeping the saliency higher. They further improvised to length adaptive Dual-CES.

3.7 Query relevance

Query relevance is the measure of finding the relationship between the searched query and the input documents. There are different techniques to find the query relevance with documents. We observe that cosine similarity is used widely to impose query relevance. For example, Roitman et al. [170] estimated the query's relevancy with summary using two similarity measures viz Bhattacharyya and cosine similarity. Other similarity measures include WordNet and NER similarities. On the other hand, the graph-based methods use the shortest path to query and query-biased ranking algorithms. For example, Canhasi [21] used inter and intra relationships between the query layer, frame layer, sentence layer, document layer, and paragraph layer to impose a query-specific ranking mechanism to the document. In contrast, few methods such as [36, 109, 118, 187] used query relevance in their submodular optimization functions. On the other hand, classical ML methods used query-dependent sentence features, and reinforcement methods imposed query reward functions. For example, Ouyang et al. [147] used query-dependent features such as word matching, semantic matching, NER matching, stop-word penalty, and sentence position to measure the query's relevance. Paper wise details are provided in Table 2.

4 EVALUATION METRICS

Evaluation metrics are used to compare different algorithms. In literature, many different evaluation metrics are used for comparing text summaries. We collected seventeen such metrics used for the evaluation of text summary. These metrics use different features/properties of the summary text in order to generate the scores. Details of how these different metrics calculate the performance of the algorithm is provided below.

ROUGE: Recall-Oriented Understudy for Gisting Evaluation [117]. It is widely used in NLP for the evaluation of summaries and translations generated automatically. ROUGE score outputs in terms of precision (P), recall (R), and F1

Table 2. Query Relevance Methods

Method	Query Relevance					
	CS[1]	SF[2]	SS[3]	QSR[4]	SO[5]	Others
Group : Semantic Analysis						
RelationListWise [216]	✓					BIR, BIN
Crowdsource and lexical analysis [135]	✓					Wordnet similarity, NER similarity
Light open-domain techniques [76]	✓					semantic triples overlap
Actor-object relationship [194]	✓	✓				
Ontology web search [163]	✓					WSQL
Transfer Learning and BM25 model [101]	✓		✓			BM25
CCTSenEmb [57]						sentence embedding models
Group : Regression						
FastSum [175]		✓				topic-title, topic-description
Semantic and syntactic feature [147], SVR [49], SVM-DBN [89]		✓				
Group : Clustering						
Combination of Agglomerative and K-means Clustering [142]	✓					
EM Clustering [12]	✓					mutual reinforcement
NMF Clustering [150]	✓		✓			
GMM [217]	✓					BIR, BIN, JS divergence
Group : Matrix Decomposition						
CLASSY [32]						query terms, POS tags, BBN's Identifinder
Dimensionality Reduction [68], Multi-facility location problem [21]	✓					
Archetypal Analysis [22]	✓					similarity graph
Synthesis matrix scientific research [69]				✓		
Group : Probabilistic methods						
Exploiting relevance, coverage and novelty [125]	✓					topic-similarity, joint probabilistic model
Contextual Query Expansion [214]			✓			correlations
S-sLDA [110]	✓					query joint distribution
Contextual Topic Model [213]						relevance language model, joint probabilistic model
Group : Submodular Optimization						
Class of Submodular Functions [118]	✓				✓	query-similarity
Coverage and textual-unit similarity [109]	✓				✓	query information coverage
MCKP [36]	✓				✓	Wordnet similarity
Event Guidance [187], Abstractive Summarization [24]	✓				✓	
Group : Reinforcement						
Simultaneous Ranking and Clustering [20]	✓					theme cluster query
Mutually reinforced manifold-ranking [19]	✓			✓		theme cluster query, mutual reinforcement
Fear the REAPER [168]						reward function
Biomedical Texts [137]	✓					
Group : Graph						
Online and Offline stage [148]						minimum spanning tree, shortest path to query
Manifold Ranking and Diversity [107]				✓		query-sensitive sentences
Global Semantic Graph [5]						concept similarity, shortest path to query
Two-layer Graph [116]	✓			✓		
Single layer Hypergraph [207]	✓			✓		
Bipartite graph + hypergraph [224]						inter-connection links
Double-Hypergraph [18]						theme cluster query, affinity propagation
Semantic Content word expansion [1]	✓					Wordnet similarity
Five-layered Graph [21]						query layer, inter-connection links
Group : Encoder-Decoder						
Knowledge Graph [48]						knowledge graph
Document QA and QMDS [114]			✓			pretrained DQA model
Coarse-to-Fine [209]	✓		✓			relevance estimator, evidence estimator
CAiRE-COVID [186]	✓					query with BART model
Seq2seq Attention model [8]	✓					TF-IDF, Word2Vec
WSL-DS [102], PreQFAS [103]						finetune queries, BERTSUM
LQSum [210]						latent query model
Group : Others						
Deep learning model [228]						query weight initializations, query penalty
Big Data [189]						URLs with query, Google Search API
Cross-Entropy method [50]		✓				
Dual-Cascade Learning [170]	✓					Bhattacharyya similarity

[1] Cosine Similarity [2] Sentence Features [3] Semantic Similarity [4] Query Specific Ranking [5] Submodular Optimization

(F) scores. In simpler words, precision signifies the percentage of results relevant to the user, whereas recall signifies total correctly classified relevant results. These metrics are used in different situations according to their needs. Out of these, F1 score is more informative to describe a model's performance in imbalanced data. ROUGE score consists of five evaluation metrics as below:

(1) ROUGE-N: It measures the overlap between the n-grams present in the reference summary (RS) and the model generated summary (S). ROUGE-N score is computed as ROUGE-N $= \frac{\sum_{S\in\{RS\}}\sum_{gram_n\in S} Count_{match}(gram_n)}{\sum_{S\in\{\mathrm{RS}\}}\sum_{gram_n\in S} Count(gram_n)}$

where, $gram_n$ represents n-gram sequences, RS signifies reference summaries and $Count_{match}$ calculates maximum number of n-grams co-occurring between a candidate and set of reference summaries.

(2) ROUGE-L: It measures the overlap of longest co-occurring of n-grams between the RS and S. $R_{lcs} = \frac{LCS(X,Y)}{m}$, $P_{lcs} = \frac{LCS(X,Y)}{n}$, and $F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}}$, where X and Y represents sentences.

(3) ROUGE-W: It is an extension of ROGUE-L with weights for evaluating consecutive LCS in the sequence. $R_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(m)}\right)$, $P_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(n)}\right)$ and $F_{wlcs} = \frac{(1+\beta^2)R_{wlcs}P_{wlcs}}{R_{wlcs}+\beta^2 P_{wlcs}}$.

(4) ROUGE-S: It measures the overlap of co-occurrence of skip-bigrams [75] in the reference and system-generated summaries. $R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)}$, $P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)}$, $F_{skip2} = \frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2}+\beta^2 P_{skip2}}$.

(5) ROUGE-SU: It is an extension of ROGUE-S with unigram co-occurrence.

Bilingual Evaluation Understudy (BLEU) [149]. BLEU metric measures how close are the word choices between the generated summary and human referenced summary. Summaries with sentences having a higher number of matches give a higher BLEU score. The range of the BLEU score lies between 0 to 1. The limitation is that it checks for exact matching between the n-grams. Hence, less preferred in abstractive summarization.

METEOR [104]. METEOR fix the limitations shown in BLEU metric by computing the harmonic mean of unigrams with precision and recall values. Initially, it only performs exact, stem, and synonym matching between the sentences, later, 'METEOR Universal' [37] performs paraphrase matching along with previous matching between the pairs of sentences. Meteor score ranges from 0 to 1 and a higher score represents a better hypothesis.

Pyramid and Responsiveness. These are the manual metrics used in TAC[1] dataset. Pyramid is evaluated on the popularity of information shared across the gold summaries. The information shared across different gold-standards capture higher weights in the generated summary. On the other hand, responsiveness metric measures to what extent generated summary satisfies the use query. There is no involvement of gold summary in measurement of responsiveness metric.

AutoSummENG [62]. AutoSummENG uses n-gram graph representation for evaluation. It uses statistical methods to extract the relation between the n-grams. These relations are used to draw a graph with edge weights as the mean distance between adjacent n-grams. The comparison is based on character and word n-gram representation. The similarity between the graphs is computed in two ways, using (a) isomorphism and (b) edit-distance. Along with graphs, they also explored histograms representation which gives better results with word n-grams.

BEwT-E (BE with Transformations for Evaluation) [193]. The previous metrics cannot handle sentences with alternate phrasal tokens and multi-word names or name aliases. 'BEwT-E' measures expressive syntactic units called Basic Elements (BE) between the summaries. Certain weights are also associated with these BEs that contain essential contents such as root, total, or binary tallying. They performed several transformations to match the contents that are semantically similar but lexically different. It uses a successive shortest path algorithm to compute the optimal BE matching possible from various transformations.

CIDEr [196]. CIDEr measures the similarity between generated and human-ground truth summaries. This metric is comparably more correlated to the human annotation consensus. They form triplet annotations for each input-one reference summary and two candidate summaries. The objective is to choose which candidate sentences are more

[1] https://tac.nist.gov/

similar to a maximum number of references. CIDEr is calculated as $\text{CIDEr}_n(c_i, S_i) = \frac{1}{m}\sum_j \frac{g^n(c_i)\cdot g^n(s_{ij})}{\|g^n(c_i)\|\|g^n(s_{ij})\|}$ where, $g^n(c_i)$ represent n-gram vector of $g_k(c_i)$ that signifies TF-IDF representation.

CHRF [157]. CHRF calculates character n-gram F-score between the candidate and reference sentence. $CHRF\beta = \left(1+\beta^2\right)\frac{\text{CHRP}\cdot\text{CHRR}}{\beta^2\cdot\text{CHRP}+\text{CHRR}}$ is the formula for computing CHRF, where, CHRP and CHRR represent the character n-gram precision and recall, respectively, and β parameter gives more importance to recall values than the precision.

ROUGE-WE [144]. The original ROUGE only consider lexical similarities and unsuitable for abstractive summarization. 'ROUGE-WE' uses pre-trained word-embeddings for computing the overlap between the sentences. ROUGE-WE $f_{WE}(w_1, w_2) = 0$ if v_1 or v_2 are out-of-vocabulary words, else, $v_1 \cdot v_2$, where v_i represents the word-embeddings of the unigram w_i.

S3 [156]. S3 leverage the advantages of various metrics by combining their strategies and learned using regression to get the best combination of features. They define two types of correlation viz system-level and summary-level. System-level correlation learns correlation between two aggregated scored lists, while summary-level correlation learns between human judgments and candidate system scores. In our comparative study, we used S3-pyramid and S3-responsiveness for evaluation.

MoverScore [223]. MoverScore use contextualized word embeddings generated from large pre-trained models fine-tuned on various natural language inference datasets to yield better embeddings. MoverScore has two variants, viz word mover and sentence mover. Word Mover's Distance (WMD) [97] semantically aligns the most similar words together to determine the correct flow of meaning in words.

Sentence Mover's Similarity [29]. WMD fails with group of words and longer documents. Sentence Mover's Similarity is a modified version of WMD which is computed by minimizing this distance to move similar words leveraging the concepts of BOW and word embeddings. They come up with two variants viz sentence level (SMS) and sentence + word level (S+WMS). By using S+WMS, a sentence embedding can also be mapped to a word embedding. For example, "Ram is having a lot of fun." maps to "enjoy". It can also be used as a reward function in reinforcement learning to train a generative model.

BLANC [195]. BLANC evaluate the candidate summary without using any human reference summary. There are two versions of BLANC viz BLANC-help and BLANC-tune. The former identifies how well the generated summary help to reconstruct the masked tokens when passed to a model with input documents, whereas, the latter tune the model with the summary. This way, they compute the difference in accuracies achieved between finetuning and without finetuning the model. $BLANC_{help} = A_s - A_f = \frac{S_{01}-S_{10}}{S_{total}}$, where, A_s and A_f signifies accuracies with summary and filler respectively. In S_{ij}, i signifies filler while j signifies summary and 0 or 1 signifies their successful and unsuccessful unmasking. Hence, S_{01} represent count of successful summary, S_{10} represent count of successful fillers, and S_{total} is summation of S_{00}, S_{01}, S_{10}, S_{11}.

BERTScore [220]. BERTScore solves the pitfalls of previously used metrics by computing the semantic similarity between the sentences using contextual embeddings to capture the distant dependencies between the terms. The contextual embeddings produce different vector representations for the same word depending on its neighboring words. The primary model used is WordPiece BERT [39] that handles the unknown words by splitting them into the known sequence of characters.

SUPERT (SUmmarization evaluation with Pseudo references and bERT) [58]. SUPERT is an unsupervised multi-document summarization evaluation metrics that uses BERT and SBERT to measure the semantic similarity between the input and generated summaries. The workflow involves two steps viz (a) building a pseudo reference summary from the input documents and (b) measuring the semantic similarity between the pseudo reference summary and the generated summary. Instead of cosine similarity, SUPERT uses WMD as soft word alignments. The authors have used simple and graph-based heuristics to generate pseudo summaries. The SUPERT scores can also be used as a reward to train RL based summarizers.

Anchored ROUGE [200]. We have observed how source documents and reference summaries separately play their role in evaluating the previous evaluation metrics. 'Anchored ROUGE' uses source documents along with reference summaries to evaluate the generated summary. It solves the problem of the ROUGE metric, where it suffers from the hard matching of tokens. It is named so because it anchors certain lexical items from the source to the summary. This utilization of the anchor set acts as a weightage to focus more on the links referred from source documents. $\text{ROUGE-anchored} = \frac{\sum_{ref \in RefSumm} \sum_{d \in C_{ref}} \min(T(d,peer), T(d,ref))}{\sum_{ref \in RefSumm} \sum_{d \in C_{ref}} T(d,ref)}$ where, $RefSumm$ represents collection of human reference summaries, C_{ref} represents the anchored set, $T(d, ref)$ represent the count of summary particles between the anchored set and the reference summary.

QAEval [38]. The previously discussed metrics have not included any questionnaire while evaluating the summary. 'QAEval' evaluates the content quality of the summary using question-answering (QA) pairs. The information given in the reference summary is molded into QA pairs, and the generated summary is evaluated with these QA pairs. Although its objective is similar to QMDS, it primarily focuses on nouns as answers which is insufficient to compute the complete information. It follows predicate-argument relations, so it is incapable for evaluating those sentences that do not have such relations.

Most metrics mentioned above only evaluate the summary based on the reference summary, which makes the evaluation biased. However, for evaluating the QMDS summary, one must also consider the source documents. This way, we can evaluate how well the generated summary answers the query based on the information given in the source documents. Hence, the metrics as mentioned earlier are not well-suited for QMDS tasks. They cannot evaluate the summaries based on their query-relevance, conciseness, factual correction, temporal relation and non-redundancy. Also, applying MDS metrics would not be favorable as it requires query-focused reference summaries, which are not widely available for QMDS tasks. Considering the above factors, there exists a research gap for query-focused summarization evaluation metrics independent of human-reference summaries.

5 DATASETS

There are plenty of datasets available for SDS (CNN/Daily News dataset, Debatepedia [65]), and MDS (MDSWriter [131], Multi-News [47], auto-hMDS [231], WCEP [61], Multi-XScience [124]). However, only a handful of datasets are available for QMDS. In this section, we provide a brief description of these benchmark QMDS datasets. Out of these, we have used the seven most widely used QMDS datasets in our comparative study to get an overall idea of different methods in QMDS. Table 3 shows the dataset statistics, where each column signifies their average values.

Table 3. Benchmark Dataset Statistics

Datasets	#Topics	#Docs per topic	#Sentences	#Queries	Summary Length (#words)	#Gold Summaries	Availability
DUC-2005	50	32	45,931	50	250	4	On request
DUC-2006	50	25	34,560	50	250	4	On request
DUC-2007	45	25	24,282	45	250	6	On request
TAC-2008	96	10	23,193	96	100	4	On request
TAC-2009	88	10	22,128	88	100	4	On request
TAC-2010	92	10	22,360	92	100	4	On request
TD-QFS	4	185	6152	10	250	4	Public
QMSum	7.8	232	2104194 (approx.)	7.8	69.6	1	Public
AQUAMUSE	-	6	66.4 (per input doc)	5,519	105.9	1	Public
QMDSCNN	-	(6.5/6.5/6.5) (Train/Val/Test)	-	(287,113/13,368/11,490) (Train/Val/Test)	250	1	On request
QMDSIR	-	(5.8/5.4/5.5) (Train/Val/Test)	-	(82,076/10,259/10,260) (Train/Val/Test)	250	1	On request

Document Understanding Conference (DUC) and Text Analysis Conference (TAC). National Institute of Standards and Technology (NIST) has organized DUC[2] and TAC[3] to conduct different summarization tasks varying from SDS to QFS. DUC conducted summarization competitions from 2001-2007 and later, in 2008, joined as a summarization track in TAC. Both datasets contain multiple news articles with queries covering domains such as politics, biographies, disasters, and others. TAC focused on two summarization types, i.e., update and opinion pilot. In update summarization, the user is already familiar with the topic, whereas the opinion pilot is an opinion summarization based on blogs. These datasets also include four human-curated gold standard summaries for evaluation. The task is to generate a 250 words summary for DUC and 100 words for TAC documents.

Topically Diverse Query Focus Summarization (TD-QFS). TDQFS [7] dataset includes documents related to asthma, lung cancer, obesity, and Alzheimer's, along with multiple queries referring to their causes, treatments, and others respectively. They have a controlled level of topic concentration in the documents. These queries are extracted from PubMed query logs and are much shorter than DUC queries. Similar to DUC, the task is to generate a 250 words summary of the input documents based on the given query.

QMSUM [227]. QMSum dataset is a query-based multi-domain meeting summarization dataset that consists of multiple general and domain-specific queries per document. It consists of meetings from three domains viz product, academic, and committee. In comparison to the previous datasets, the documents are longer, and the summary length varies for general (50-150 words) and specific queries (20-100 words).

Automatically Generating Datasets for Query-Based Multi-Document Summarization (AQUAMUSE) [93]. AQUAMUSE is one of the recent question-answering datasets generated using the Google Natural Questions (NQ) dataset [98] and Common Crawl corpus [160]. The inputs are matched documents from the Common Crawl corpus, whereas the query and the summary (long answers) are extracted from the NQ dataset. It provides both extractive and abstractive summaries for each query.

QMDSCNN and QMDSIR [151]. QMDSCNN is generated by restructuring the SDS dataset (CNN/Daily Mail) and the other QMDSIR by mining actual web queries from the search logs of Bing. The former has actual summaries with simulated queries, which makes the query less informative. In contrast, the latter has actual queries with simulated summaries, which may not contain the complete summary based on input documents.

We have discussed the limitations of existing QMDS datasets in Table 4.

[2]http://duc.nist.gov/
[3]http://www.nist.gov/tac/

Table 4. Limitations of existing QMDS datasets

Dataset	Limitation
DUC 2005-2007 TAC 2008-2010	1. DUC and TAC dataset suffers from excessive topic concentration. The dataset is designed as if all the sentences are relevant, hence, model won't improve much even after filtering the irrelevant words. 2. Dataset is quite smaller in size, hence, cannot be used for training ML and DL models.
TDQFS	1. The dataset is very small that makes it difficult to train large models.
QMSUM	It is multi-domain meeting summarization dataset, instead of multi-document.
AQUAMUSE	1. The input is given as website links out of which many doesn't exist anymore, that makes it difficult for a user to use it for model building. 2. The dataset set size is not too large to train large DL models.
QMDSCNN	It has real summaries with stimulated queries which makes the query less informative.
QMDSIR	It has real queries with stimulated summaries which may or may not contain the complete summary based on input documents.

6 COMPARATIVE STUDY

In order to understand how different approaches perform on the same dataset, we experimented with nine methods, two methods each from first three groups (semantic analysis, classical ML, statistic and matrix decomposition) and one method each from remaining three groups. Seven evaluation metrics are used to compare the performance. Details of the experiments and results are presented in this section.

6.1 Comparing methods

Nine algorithms used in the evaluation on eight benchmark datasets (Table 3). The methods are selected based on higher citations in the group. These are

(1) Valizadeh and Brazdil [194] (VB15) - It is a semantic analysis-based method that uses an ensemble of various models to perform QMDS. The model utilizes the human gold-standard summaries for creating the training data. The feature space consists of various query-dependent and independent sentence features.

(2) Lamsiyah et al. [101] (L21) - It is an unsupervised learning-based method that uses contextual word embeddings to compute the semantic relationships between the words to capture a better meaning of the sentence. In contrast to the original paper, where authors have used two pre-trained models USE-DAN[4] and USE-Transformer[5], here, we have experimented only with USE-DAN.

(3) Ouyang et al. [147] (O11) - It is a classical ML method. Unlike the original approach, where training was performed only with DUC datasets, we performed training using a combination of DUC and TAC datasets. For example, to evaluate DUC 2007 dataset, the model is trained on DUC 2005, 2006, and TAC 2008, 2009, 2010.

(4) Kinyanjui et al. [89] (K21) - It is also a classical ML method that consists of hybridization of SVM and DBM model. We used train-test split of 85:15 in each of the 8 datasets. The original paper only experimented for DUC 2006. In their feature space, they have a feature called temporal dimension (td) that is calculated as the inverse of the difference between the document published year and year 2000 (assumed by the authors). Since, for DUC 2005, TDQFS and QMSUM, the authors have not provided any value for temporal dimension, so we took that feature as 0.

(5) Hachey et al. [68] (H06) - This algorithm is based on statistic and matrix decomposition methods. The decomposition value is decided by calculating variances at different dimensions.

(6) Chali et al. [24] (C17) - This algorithm is also based on statistic and matrix decomposition methods. It uses a submodular function with sentence compression and merging function.

(7) Cai and Li [19] (CL12) - In this reinforcement-based method, two mutually reinforced algorithms, RDRP, and RARP are proposed to perform reinforcement during and after propagation. We have explored RDRP and used

[4] https://tfhub.dev/google/universal-sentence-encoder/4
[5] https://tfhub.dev/google/universal-sentence-encoder-qa/3

k-means clustering to identify the theme clusters in our comparative study. The optimal k-value is calculated using the elbow method.

(8) Xiong and Ji [207] (XJ16) - It is a graph-based method. In the HDP topic modeling, we consider only the top 20 topics.

(9) Laskar et al. [103] (L22) - It is a learning based method. It uses weakly supervised learning with distant supervision to generate query focused summaries. For evaluation in DUC (2005, 2006, 2007) dataset, we used two datasets for training the BERTSUM and the other for testing. In case of TAC (2008, 2009, 2010), we used the BERTSUM model trained on DUC dataset because both datasets are based on news articles. In case of TDQFS and QMSUM, we followed the 85:15 ratio for training and testing. We trained them separately because TDQFS and QMSUM are based on medical and meeting summaries, so their distribution is different from news articles (DUC and TAC). Due to memory limitations, we fine-tuned the BERTSUM model for 10 epochs with batch size 4. Due to large number of topics in case of TDQFS and QMSUM, we restricted our training to small sample of topic to meet the memory constraints.

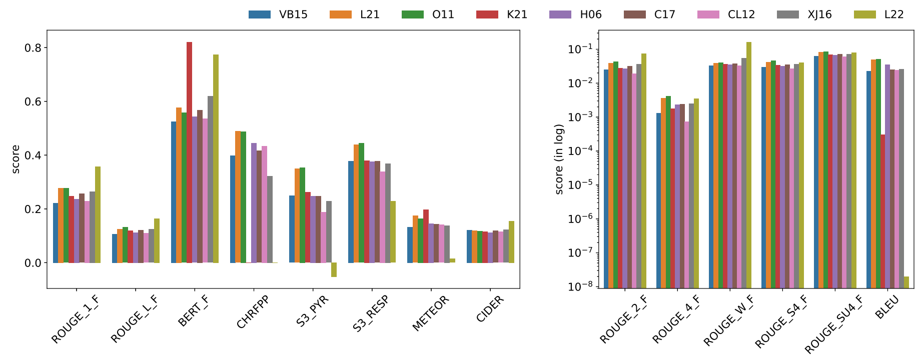
The evaluation metrics used for the comparative study are ROUGE-N measures, BERTScore, BLEU, CHRF, S3, METEOR, and CIDer. Due to space limitation, we reported only ROUGE F1 and BERTScore F1 values in the comparative results, although, in our experiments, we calculated all three, including precision and recall scores.

6.2 Results

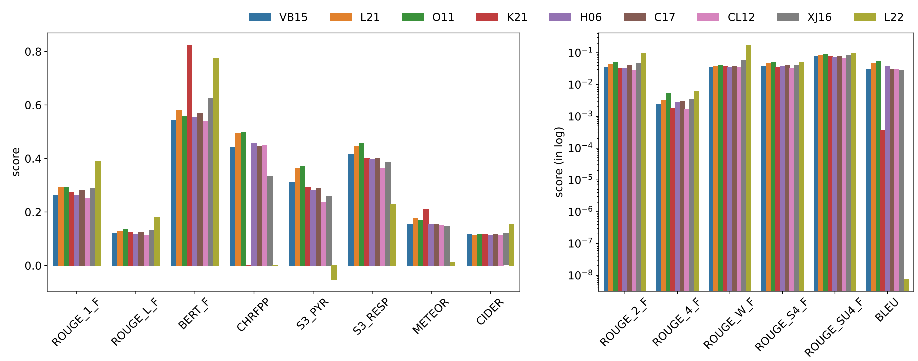
The major results of our experiments are shown in Figs. 9, 10, 11, and 12. For better understanding, we have represented some metrics scores in the logarithmic scale whose values are low. Numerical results are available in the supplementary material.

DUC 2005, 2006, and 2007. Fig. 9 shows the plot for different metrics for various methods on DUC 2005, 2006, and 2007 data sets. It is evident from Fig. 9 that K21 produces superior results compared to any other methods for BERTScore metrics, while for ROUGE scores, L22 shows the highest results for most of the cases. For example, K21 achieved 0.82111, 0.82528, and 0.82268 BERTScore F1 values for DUC 2005, 2006, and 2007 respectively, which are higher by 6%, 6.5%, and 6.75% than the nearest value of L22 (0.77402, 0.77516, and 0.77064). On the contrary, L22 gets 28%, 32%, and 21% higher scores than the nearest L21 for DUC 2005 and O11 for DUC 2006 & 2007 in terms of ROUGE-1 values. O11, K21, and L22 show higher scores than the other methods, except for the DUC 2005's CHRFPP score, where L21 provided the best result. One should note that although L21 is classified under semantic analysis-based methods, it is an unsupervised DL method that uses pre-trained embeddings in its pipeline. However, it performs poorly as compared to L22.

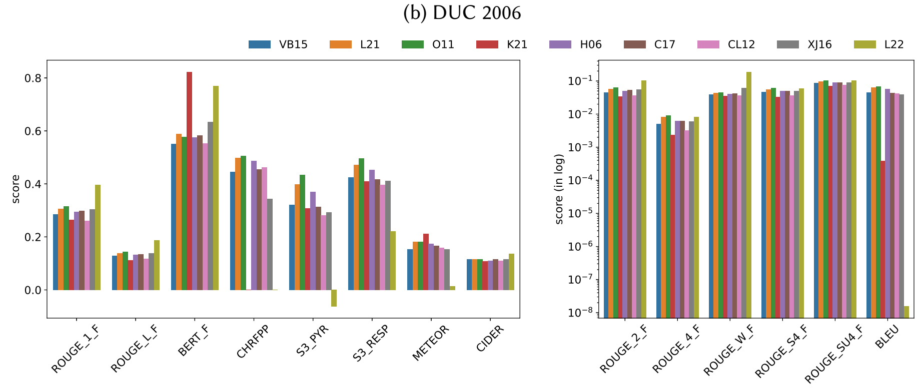
TAC 2008, 2009, and 2010. The bar chart of Fig. 10 shows the scores of different metrics for various methods on TAC 2008, 2009, and 2010 data sets. Scores of L22 are highest compared to any other methods for all ROUGE measures and CIDER metrics in all three TAC datasets except in TAC 2009, where O11 scored highest in ROUGE-4 and ROUGE-S4. For example, L22 achieved 0.09417 for the ROUGE-2 F1 value for TAC 2009, which is higher by 64% than the nearest value of O11 (0.05716). Interestingly, O11 gets 13% higher results than L22 for TAC 2009 ROUGE-4 value. Similar to DUC results, K21 performed better than others in BERTScores and METEOR, e.g., K21 gets a 5% higher BERTScore F1 score (0.83312) than the nearest L22 (0.78774) for TAC 2008 dataset. We could observe that analogous to DUC results, TAC also has comparable performance between the three methods, O11, K21, and L22. This implies how ML and DL-based methods modeled a better QMDS summarizer than others.



(a) DUC 2005



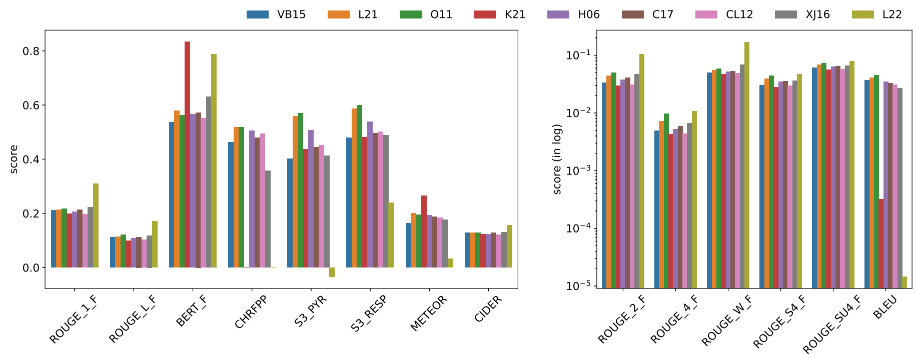
(b) DUC 2006



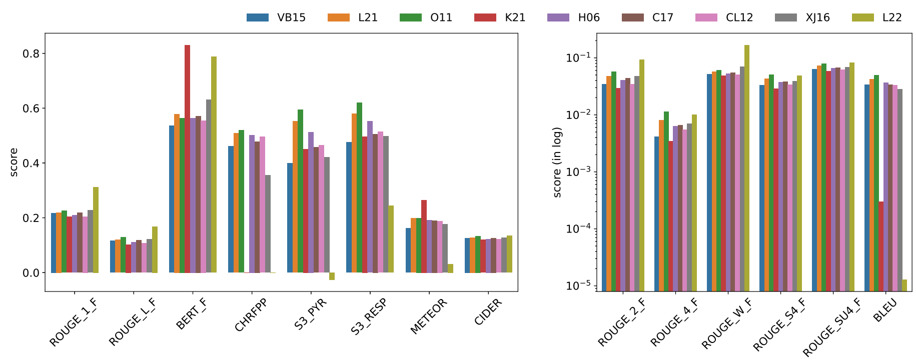
(c) DUC 2007

Fig. 9. Scores of different algorithms for different DUC dataset

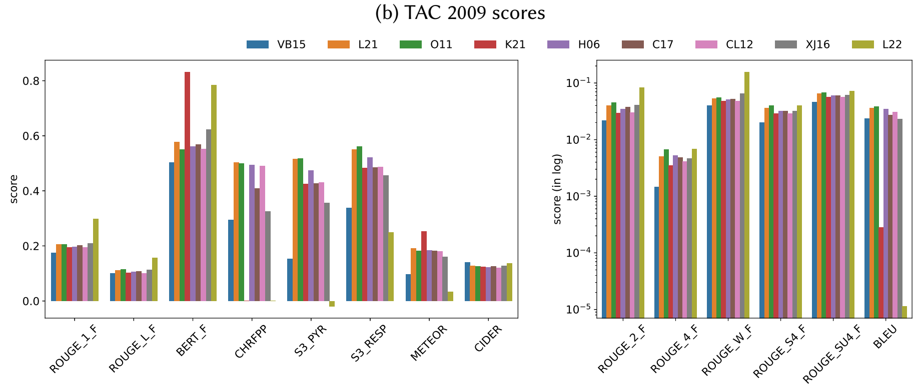
TD-QFS. The evaluation results on TD-QFS dataset are shown in Fig. 11. Unlike the previous two data sets, one can observe that L21 scored highest in almost all the ROUGE scores except for ROUGE-W where L22 scored 162% better F1 (0.13987) than L21(0.05331). Similar to DUC and TAC results, K21 performed better than others in BERTScores and METEOR, e.g., K21 gets a 4.8% higher BERTScore F1 score (0.82519) than the nearest L22 (0.78734). Unlike DUC and TAC



(a) TAC 2008 scores



(b) TAC 2009 scores



(c) TAC 2010 scores

Fig. 10. Scores of different algorithms for different TAC dataset

datasets, L21 scored the highest in BLEU, CHRFPP, S3_PYR, S3_RESP, e.g., L21 gets a 42% higher BLEU score (0.14416) than the nearest K21 (0.10133). This is because the queries are shorter in length than the DUC datasets, and we discussed before that regression methods use query-dependent features. Hence, K21 slightly performed low in comparison to L21.

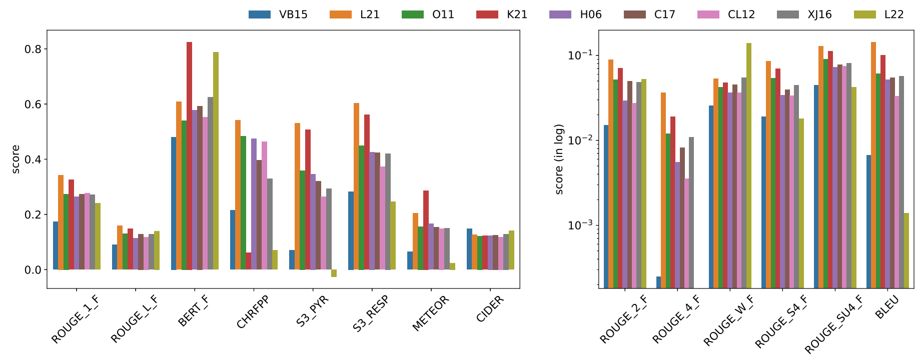


Fig. 11. TD-QFS scores

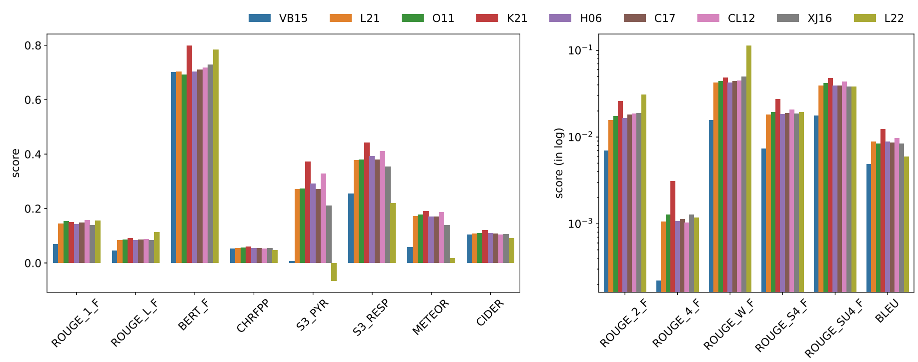


Fig. 12. QMSum scores

We have also mentioned earlier that L22 was trained only for small sample of topics which ultimately degraded its performance.

QMSUM. The evaluation results on QMSUM dataset are shown in Fig. 12. Unlike the previous data sets, one can observe that K21 scored highest in almost all the metrics except for ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-W F1. L22 scored 116% higher in ROUGE-W F1 (0.10462) as compared to K21 (0.04835).

One could observe that L22 scored the highest ROUGE-W F1 in all eight datasets. This implies that L22 effectively adds essential phrases in the final summary. An interesting observation of our study is how DL methods are improving over time. It is evident from the results of L21, O11, K21, and L22. While L21 produces lower scores than regression-based methods O11 and K21, L22 shows higher scores for at least six different metrics, including ROUGE-1, ROUGE-2, ROUGE-4, ROUGE-L, ROUGE-W, and CIDER.

7 CONCLUSION

We presented the first systematic review of the various methods used for Query-focused Multi-Document Summarization. We have classified different methodologies into six different groups based on the similarities of their text summarization technique. Along with that, we also discussed the recent developments in MDS. Further, we curated a list of 17 metrics that are used for evaluating text summarization algorithms.

We reported a detailed comparative study between the six groups classified here over eight QMDS datasets. This study shows that DL and classical ML methods performed the better than other methodologies developed for QMDS. Our analysis also reveal that the DL methods are improving over time. Although, we found that the data sets available for QMDS is highly limited in their size to train large DL models, our analysis highlights that the state-of-the-art DL based method performs better than other methods. There are many large scale data sets for SDS and MDS; large scale data set, if developed for QMDS, that can provide further improved results. In-fact larger DL models can be trained with large size data.

QMDS is still relatively unexplored compared to other variants of text summarization. The study identified following four major challenges in the research of QMDS.

7.1 Challenges

(1) **Unavailability of Large Data Set:** The available benchmark datasets such as DUC, TAC, TDQFS, and others have a relatively minor number of samples for training neural network models. It is crucial to develop high-quality datasets for QMDS that consist of rich, diversified documents with lower extractive biases.
(2) **Solutions Available in Limited Context:** The current QMDS datasets mainly consist of documents from the news (DUC, TAC) and medical (TDQFS) domains. Hence the available literature only provides solutions in the context of news and medical documents. However, there are other domains where query-based summarization is required. Such use-cases include answering legal and financial queries, summarizing conversational documents, and recommendation/review summarization.
(3) **Unavailability QMDS Specific Evaluation Metrics:** The QMDS evaluation metric should reflect the following properties - a) evaluation of the cross-document relations between the input and the generated summary, b) a measure to identify how completely the summary answers the query based on the given input documents, c) a measure to calculate redundancy of information in the output summary, d) a measure to evaluate fluency, consistency, a factual correction, and coherency. Although we discussed seventeen evaluation metrics in Section 4 for text summarization, none are explicitly developed for QMDS problems addressing the above four points. Thus, the unavailability of the correct QMDS evaluation metric makes it quite challenging to measure the performance of generated summary.
(4) **Different type of Queries:** The QMDS system should be robust to questions, multi-entity-based queries, longer queries combining multiple sub-queries, and others. In order to solve such challenges, research is going on to redesign proxy queries and re-training system components; however, they could be more computationally efficient and infeasible after model deployment [210]. Hence, we need robust models to handle or redesign such queries in a generic form for our models to process.

7.2 Future Directions

The literature on semantic analysis in NLP is rich [71, 91, 105, 154, 212]. Many of the recent development therein could be helpful in QMDS tasks. For example, sentence representations in InferSent [31] that classify encoded vectors into entailment, contradiction, and neutral can help generate more semantically entailed summaries relating to the queries. In contrast, SBERT [166], a modified version of BERT with siamese and triplet networks, creates semantically meaningful fixed-size sentence vectors. SBERT is computationally efficient, enabling it to summarize queries in real-life applications. Sent2vec is an unsupervised model for generating sentence embedding vectors, including sentimental semantics [134]. Sent2vec can be used in massive reviews summarization where query-focused summary with sentiment analysis is necessary. Textual entailment recognition (TER) [178] checks the direction relationship if one text fragment can entail the truth of another text fragment. In QMDS, using TER, we can eliminate redundant sentences or expressions if they entails other text fragments in summary. Explainability is studied in SDS [198] but has yet to be explored in QMDS. Recent research methods used for explainability in summarizations include attention distribution, source attribution approach, and others [146]. Future research on QMDS should incorporate such qualitative analysis for their models. Adversarial perturbations can be utilized to improve model robustness for different tasks. Zhang et al. [221] experimented with MDS datasets, including DUC2003, DUC2004, and Gigaword. Although there has been substantial research on adversarial robustness for NLP models, there needs to be more research on the robustness of QMDS models. Hence, more research is needed to propose new adversarial attacks for QMDS models. Multi-modal systems have various applications and help combine text with image, video, or audio. Meeting summarizations or news telecasts could help improvise QMDS as it could take the context from multiple modalities viz visual expressions, voice, and text. Deep learning models such as ICCN [188], MDREA [218], VisualBERT [112], UNITER [27], and others with a larger capacity to handle rich modalities in QMDS are needed. Multi-modal QMDS has been largely unexplored and has future applications.

REFERENCES

[1] Asad Abdi, Norisma Idris, Ramiz Aliguliyev, and Rasim Alguliyev. 2017. Query-based multi-documents summarization using linguistic knowledge and content word expansion. *Soft Computing* (04 2017), 1–17.

[2] Raksha Agarwal, Niladri Chatterjee, David Pinto, Beatriz Beltrán, and Vivek Singh. 2022. Query-Focused Multi-Document Text Summarization Using Fuzzy Inference. *Journal of Intelligent 'I&' Fuzzy Systems* 42 (2022), 4641–4652.

[3] Amanuel Alambo, Cori Lohstroh, Erik Madaus, Swati Padhee, Brandy Foster, Tanvi Banerjee, Krishnaprasad Thirunarayan, and Michael Raymer. 2020. Topic-Centric Unsupervised Multi-Document Summarization of Scientific and News Articles. *Proceedings - IEEE International Conference on Big Data* (2020), 591–596.

[4] Aysa Siddika Asa, Sumya Akter, Md Palash Uddin, Md Delowar Hossain, Shikhor Kumer Roy, and Masud Ibn Afjal. 2017. A Comprehensive Survey on Extractive Text Summarization Techniques. *AJER* 6 (2017), 226–239.

[5] J Balaji, TV Geetha, and Ranjani Parthasarathi. 2014. A Graph based query focused multi-document summarization. *International Journal of Intelligent Information Technologies (IJIIT)* 10, 1 (2014), 16–41.

[6] Leonard E. Baum and Ted Petrie. 1966. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics* 37, 6 (1966), 1554 – 1563.

[7] Tal Baumel, Raphael Cohen, and Michael Elhadad. 2016. Topic Concentration in Query Focused Summarization Datasets. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, Arizona). 2573–2579.

[8] Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query Focused Abstractive Summarization: Incorporating Query Relevance, Multi-Document Coverage, and Summary Length Constraints into seq2seq Models.

[9] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[10] Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly Learning to Extract and Compress. In *Proceedings of ACL: HLT* (Portland, Oregon). 481–490.

[11] Mrunal S Bewoor and Suhas H Patil. 2018. Empirical analysis of single and multi document summarization using clustering algorithms. *Engineering, Technology & Applied Science Research* 8, 1 (2018), 2562–2567.

[12] Kalyani Bhagat and MD Ingle. 2014. Multi document summarization using EM Clustering. *International organization of Scientific Research Journal of Engineering (IOSRJEN)* 4, 05 (2014), 45–50.

[13] Mohammad Bidoki, Mohammad R. Moosavi, and Mostafa Fakhrahmad. 2020. A semantic approach to extractive multi-document summarization: Applying sentence expansion for tuning of conceptual densities. *Information Processing and Management* 57, 6 (2020), 102341.

[14] Chris Biemann. 2006. Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*. ACL, New York City, 73–80.

[15] David M. Blei and Jon D. McAuliffe. 2007. Supervised Topic Models. In *Proceedings of NeurIPS* (Vancouver, Canada). 121–128.

[16] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[17] Alexei Borodin. 2009. Determinantal point processes. *arXiv preprint arXiv:0911.1153* (2009).

[18] Xiaoyan Cai, Junwei Han, Lei Guo, and Libin Yang. 2016. Double-Hypergraph Based Sentence Ranking for Query-Focused Multi-document Summarizaton. In *IEEE Web Intelligence Workshops (WIW)* (Melbourne, Australia). 112–118.

[19] Xiaoyan Cai and Wenjie Li. 2012. Mutually Reinforced Manifold-Ranking Based Relevance Propagation Model for Query-Focused Multi-Document Summarization. *IEEE - TASLP* 20 (2012), 1597–1607.

[20] Xiaoyan Cai, Wenjie Li, You Ouyang, and Hong Yan. 2010. Simultaneous Ranking and Clustering of Sentences: A Reinforcement Approach to Multi-Document Summarization. In *Proceedings of COLING 2010*. Beijing, 134–142.

[21] Ercan Canhasi. 2017. Query Focused Multi document Summarization Based on the Multi facility Location Problem. In *Computer Science On-line Conference*. Springer, 210–219.

[22] Ercan Canhasi and Igor Kononenko. 2016. Automatic Extractive Multi-document Summarization Based on Archetypal Analysis. In *Non-negative Matrix Factorization Techniques*. Springer, 75–88.

[23] Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Australia). 335–336.

[24] Yllias Chali, Moin Tanvee, and Mir Tafseer Nayeem. 2017. Towards abstractive multi-document summarization using submodular function-based framework, sentence compression and merging. In *Proceedings of IJCNLP*. Taipei, 418–424.

[25] Moye Chen, Wei Li, Jiachen Liu, Xinyan Xiao, Hua Wu, and Haifeng Wang. 2021. SgSum:Transforming Multi-document Summarization into Sub-graph Selection. In *Proceedings of EMNLP*. 4063–4074.

[26] Yen-Chun Chen and Mohit Bansal. 2018. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In *ACL* (Melbourne, Australia). 675–686.

[27] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: UNiversal Image-TExt Representation Learning. In *ECCV 2020* (Glasgow, United Kingdom). 104–120.

[28] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of EMNLP*. Doha, 1724–1734.

[29] Elizabeth Clark, Asli Celikyilmaz, and Noah A Smith. 2019. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of ACL* (Florence, Italy). 2748–2760.

[30] David Cohn and Huan Chang. 2000. Learning to Probabilistically Identify Authoritative Documents. In *ICML*. San Francisco, 167–174.

[31] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of EMNLP*. Copenhagen, 670–680.

[32] John M Conroy, Judith D Schlesinger, and Jade Goldstein Stewart. 2005. CLASSY query-based multi-document summarization. In *Proceedings of the 2005 Document Understanding Workshop, Boston*. Citeseer.

[33] W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search engines: Information retrieval in practice*. Vol. 520. Addison-Wesley Reading.

[34] Peng Cui and Le Hu. 2021. Topic-Guided Abstractive Multi-Document Summarization. In *Proceedings of EMNLP*. Punta Cana, 1463–1472.

[35] Adele Cutler and Leo Breiman. 1994. Archetypal Analysis. *Technometrics* 36, 4 (1994), 338–347.

[36] Fatemeh Ghiyafeh Davoodi and Yllias Chali. 2015. Semi-extractive Multi-document Summarization via Submodular Functions. In *International Conference on Statistical Language and Speech Processing*. Springer, Cham, 96–110.

[37] Michael Denkowski and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Statistical Machine Translation*. Baltimore, 376–380.

[38] Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2021. Towards Question-Answering as an Automatic Metric for Evaluating the Content Quality of a Summary. *Transactions of ACL* 9 (2021), 774–789.

[39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL:HLT*. Minneapolis, Minnesota, 4171–4186.

[40] Apurva D Dhawale, Sonali B Kulkarni, and Vaishali Kumbhakarna. 2019. Survey of Progressive Era of Text Summarization for Indian and Foreign Languages Using Natural Language Processing. In *International Conference on Innovative Data Communication Technologies and Application*. Springer, 654–662.

[41] William B. Dolan and Chris Brockett. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

[42] Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications* 165 (2021), 113679.

[43] Sherif Elfayoumy and Jenny Thoppil. 2014. A survey of unstructured text summarization techniques. *International Journal of Advanced Computer Science and Applications* 5, 4 (2014), 149–154.

[44] Hassan A Elmadany, Marco Alfonse, and Mostafa Aref. 2015. XML summarization: A survey. In *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*. IEEE, 537–541.

[45] Hady Elsahar, Maximin Coavoux, Matthias Gallé, and Jos Rozen. 2021. Self-supervised and controlled multi-document opinion summarization. In *Proceedings of EACL*. Online, 1646–1662.

[46] Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research* 22 (2004), 457–479.

[47] Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Proceedings of ACL*. Florence, Italy, 1074–1084.

[48] Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale seq2seq models to multi-document inputs. *arXiv preprint arXiv:1910.08435* (2019).

[49] Aris Fanani, Yuniar Farida, Putra Arhandi, M. Mahaputra Hidayat, Abdul Muhid, and Billy Montolalu. 2019. Regression model focused on query for multi documents summarization based on significance of the sentence position. *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 17 (12 2019), 3050.

[50] Guy Feigenblat, Haggai Roitman, Odellia Boni, and David Konopnicki. 2017. Unsupervised query-focused multi-document summarization using the cross entropy method. In *ACM SIGIR* (Shinjuku, Japan). 961–964.

[51] Christiane Fellbaum and George A Miller. 2003. Morphosemantic links in WordNet. *Traitement automatique de langue* 44, 2 (2003), 69–80.

[52] Katja Filippova. 2010. Multi-Sentence Compression: Finding Shortest Paths in Word Graphs. In *Proceedings of COLING 2010*. Beijing, 322–330.

[53] Allen Institute for AI. 2021. Semantic Scholar. https://www.semanticscholar.org/. Accessed: 1 Nov 2021.

[54] E. Forgy. 1965. Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification. *Biometrics* 21 (1965), 768–769.

[55] Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review* 47, 1 (2017), 1–66.

[56] Shen Gao, Xiuying Chen, Zhaochun Ren, Dongyan Zhao, and Rui Yan. 2020. From standard summarization to new tasks and beyond: Summarization with manifold information. *arXiv preprint arXiv:2005.04684* (2020).

[57] Yang Gao, Yue Xu, Heyan Huang, Qian Liu, Linjing Wei, and Luyang Liu. 2020. Jointly Learning Topics in Sentence Embedding for Document Summarization. *IEEE Transactions on Knowledge and Data Engineering* 32, 4 (2020), 688–699.

[58] Yang Gao, Wei Zhao, and Steffen Eger. 2020. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. *arXiv preprint arXiv:2005.03724* (2020).

[59] Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-Up Abstractive Summarization. In *Proceedings of EMNLP*. Brussels, 4098–4109.

[60] Saeedeh Gholamrezazadeh, Mohsen Amini Salehi, and Bahareh Gholamzadeh. 2009. A comprehensive survey on text summarization systems. In *2009 2nd International Conference on Computer Science and its Applications*. IEEE, 1–6.

[61] Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A Large-Scale Multi-Document Summarization Dataset from the Wikipedia Current Events Portal. In *Proceedings of ACL*. Online, 1302–1308.

[62] George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. 2008. Summarization system evaluation revisited: N-gram graphs. *ACM TSLP* 5, 3 (2008), 1–39.

[63] G. Golub and W. Kahan. 1965. Calculating the Singular Values and Pseudo-Inverse of a Matrix. *Siam Journal on Numerical Analysis* 2 (01 1965), 205–224.

[64] Gene Golub and Charles Loan. 1996. *Matrix Computations, 3rd ed.* Johns Hopkins University Press, USA.

[65] Swapna Gottipati, Minghui Qiu, Yanchuan Sim, Jing Jiang, and Noah A. Smith. 2013. Learning Topics and Positions from Debatepedia. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. ACL, Seattle, Washington, USA, 1858–1868.

[66] Aaryan Gupta, Inder Khatri, et al. 2020. A Review on Various Techniques of Automatic Text Summarization. In *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 1379–1384.

[67] Surabhi Gupta, Ani Nenkova, and Dan Jurafsky. 2007. Measuring Importance and Query Relevance in Topic-focused Multi-document Summarization. In *Proceedings of ACL*. Prague, 193–196.

[68] Ben Hachey, Gabriel Murray, and David Reitter. 2006. Dimensionality reduction aids term co-occurrence based multi-document summarization. In *Workshop on task-focused summarization and question answering* (Sydney). 1–7.

[69] Hayato Hashimoto, Kazutoshi Shinoda, Hikaru Yokono, and Akiko Aizawa. 2017. Automatic Generation of Review Matrices as Multi-document Summarization of Scientific Papers.. In *Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries @ SIGIR (1)*. 69–82.

[70] Tingting He, Wei Shao, HuaSong Xiao, and Po Hu. 2007. The implementation of a query-directed multi-document summarization system. In *Sixth International Conference on Advanced Language Processing and Web Information Technology (ALPIT 2007)*. IEEE, 105–110.

[71] Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning Distributed Representations of Sentences from Unlabelled Data. In *Proceedings of NAACL: HLT*. San Diego, 1367–1377.

[72] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation* 9 (12 1997), 1735–80.

[73] Thomas Hofmann. 2013. Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705* (2013).

[74] Chris Hokamp, Demian Gholipour Ghalandari, Nghia The Pham, and John Glover. 2020. DynE: Dynamic Ensemble Decoding for Multi-Document Summarization. (2020).

[75] Xuedong Huang, Fil Alleva, Hsiao-Wuen Hon, Mei-Yuh Hwang, and Ronald Rosenfeld. 1992. An overview of the SPHINX-II speech recognition system. *Computer Speech & Language* 7 (05 1992).

[76] Quinsulon Israel, Hyoil Han, and Il-Yeol Song. 2015. Semantic analysis for focused multi-document summarization (fMDS) of text. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. New York, 339–344.

[77] J. Jagadeesh, Prasad Pingali, and Vasudeva Varma. 2007. Capturing Sentence Prior for Query-Based Multi-Document Summarization. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*. Le Centre De Hautes Etudes Internationales's D'Informatique Documentaire, Paris, France, 798–809.

[78] Prabhudas Janjanam and CH Pradeep Reddy. 2019. Text summarization: an essential study. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*. IEEE, 1–6.

[79] Hanqi Jin and Xiaojun Wan. 2020. Abstractive multi-document summarization via joint learning with single-document summarization. In *Proceedings of EMNLP 2020*. Online, 2545–2554.

[80] Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Multi-Granularity Interaction Network for Extractive and Abstractive Multi-Document Summarization. In *Proceedings of ACL*. 6244–6254.

[81] Akanksha Joshi, Eduardo Fidalgo, Enrique Alegre, and Laura Fernández-Robles. 2019. SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications* 129 (2019), 200–215.

[82] Prashant D Joshi, MS Bewoor, and SH Patil. 2011. System for document summarization using graphs in text mining. *International Journal of Advances in Engineering & Technology* 1, 4 (2011), 204.

[83] Kastriot Kadriu and Milenko Obradovic. 2021. Extractive approach for text summarisation using graphs. *arXiv preprint arXiv:2106.10955* (2021).

[84] Jagadish S Kallimani et al. 2018. Survey on Extractive Text Summarization Methods with Multi-Document Datasets. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2113–2119.

[85] NR Kasture, Neha Yargal, Neha Nityanand Singh, Neha Kulkarni, and Vijay Mathur. 2014. A survey on methods of abstractive text summarization. *IJREST* 1, 6 (2014), 53–57.

[86] Manpreet Kaur and Dipti Srivastava. 2019. Text Summarization using Partial Textual Entailment based Graphs. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. IEEE, 366–374.

[87] Zeynab Khaleghi, Mohammad Fakhredanesh, and Maryam Hourali. 2021. MSCSO: Extractive Multi-document Summarization Based on a New Criterion of Sentences Overlapping. *Iranian Journal of Science and Technology - Transactions of Electrical Engineering* 45, 1 (2021), 195–205.

[88] Christopher S. G. Khoo, Shiyan Ou, and Dion Hoe-Lian Goh. 2002. A Hierarchical Framework for Multi-Document Summarization of Dissertation Abstracts. In *Proceedings of the 5th ICADL '02*. Springer-Verlag, Berlin, 99–110.

[89] Karari Kinyanjui, Malanga Ndenga, and H O Nyongesa. 2021. Hybridization of DBN with SVM and its Impact on Performance in Multi-Document Summarization. *Machine Learning and Applications: An International Journal* 8, 3 (2021), 37–51.

[90] Mahira Kirmani, Nida Manzoor Hakak, Mudasir Mohd, and Mohsin Mohd. 2019. Hybrid text summarization: a survey. In *Soft Computing: Theories and Applications*. Springer, 63–73.

[91] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems* 28 (2015).

[92] Dan Klein and Christopher D. Manning. 2002. Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Proceedings of NeurIPS*. Cambridge, 3–10.

[93] Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. 2020. AQuaMuSe: Automatically Generating Datasets for Query-Based Multi-Document Summarization. arXiv:2010.12694 [cs.CL]

[94] Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. 2021. CoMSum and SIBERT: A Dataset and Neural Model for Query-Based Multi-Document Summarization. In *Proceedings of ICDAR* (Lausanne, Switzerland). 84–98.

[95] Yogan Jaya Kumar and Naomie Salim. 2011. Automatic Multi Document Summarization Approaches. *Journal of Computer Science* 8, 1 (Nov. 2011), 133–140.

[96] Sheena Kurian and Sheena Mathew. 2020. Survey of Scientific Document Summarization Techniques. *Computer Science* 21, 2 (2020).

[97] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of ICML* (Lille, France). 957–966.

[98] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the ACL* 7 (2019), 453–466.

[99] MVPT Lakshika, HA Caldera, and WV Welgama. 2020. Abstractive Web News Summarization Using Knowledge Graphs. In *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)* (Colombo, Sri Lanka). IEEE, 300–301.

[100] Pedro Lamberti and Ana Majtey. 2003. Non-logarithmic Jensen–Shannon divergence. *Physica A: Statistical Mechanics and its Applications* 329 (11 2003), 81–90.

[101] Salima Lamsiyah, Abdelkader El Mahdaouy, Said Ouatik El Alaoui, and Bernard Espinasse. 2021. Unsupervised query-focused multi-document summarization based on transfer learning from sentence embedding models, BM25 model, and maximal marginal relevance criterion. *Journal of Ambient Intelligence and Humanized Computing* 14, 3 (2021), 1–18.

[102] Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang. 2020. WSL-DS: Weakly Supervised Learning with Distant Supervision for Query Focused Multi-Document Abstractive Summarization. In *Proceedings of COLING*. Barcelona (Online), 5647–5654.

[103] Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang. 2022. Domain Adaptation with Pre-trained Transformers for Query Focused Abstractive Text Summarization. *Computational Linguistics* 48, 2 (2022), 1–42.

[104] Alon Lavie and Abhaya Agarwal. 2007. Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proceedings of ACL on Statistical Machine Translation* (Prague). 228–231.

[105] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of ICML*. 1188–1196.

[106] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.

[107] Kai Lei and Yi Fan Zeng. 2013. A Novel Biased Diversity Ranking Model for Query-Oriented Multi-document Summarization. In *Applied Mechanics and Materials*. Trans Tech Publications Ltd.

[108] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint* (2019).

[109] Jingxuan Li, Lei Li, and Tao Li. 2012. Multi-document summarization via submodularity. *Applied Intelligence* 37, 3 (2012), 420–430.

[110] Jiwei Li and Sujian Li. 2013. A novel feature-based bayesian model for query focused multi-document summarization. *Transactions of ACL* 1 (2013), 89–98.

[111] Jing Li, Le Sun, Chunyu Kit, and Jonathan Webster. 2007. A query-focused multi-document summarizer based on lexical chains. In *Proceedings of DUC-2007*. 29.

[112] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557* (2019).

[113] Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging Graph to Improve Abstractive Multi-Document Summarization. In *Proceedings of ACL*. 6232–6243.

[114] Weikang Li, Xingxing Zhang, Yunfang Wu, Furu Wei, and Ming Zhou. 2019. Document-Based Question Answering Improves Query-Focused Multi-document Summarization. In *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 41–52.

[115] Wei Li and Hai Zhuge. 2021. Abstractive multi-document summarization based on semantic link network. *IEEE Transactions on Knowledge and Data Engineering* 33, 1 (2021), 43–54.

[116] Yanran Li and Sujian Li. 2014. Query-focused Multi-Document Summarization: Combining a Topic Model with Graph-based Semi-supervised Learning. In *Proceedings of COLING 2014*. Dublin, 1197–1207.

[117] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of summaries. *Proceedings of ACL Workshop: Text Summarization Branches Out*, 10.

[118] Hui Lin and Jeff Bilmes. 2011. A Class of Submodular Functions for Document Summarization. In *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies*. ACL, Portland, Oregon, USA, 510–520.

[119] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198* (2018).

[120] Xiaohua Liu, Yitong Li, Furu Wei, and Ming Zhou. 2012. Graph-based multi-tweet summarization using social signals. In *Proceedings of COLING 2012* (Mumbai). 1699–1714.

[121] Yang Liu. 2019. Fine-tune BERT for extractive summarization. *arXiv preprint arXiv:1903.10318* (2019).

[122] Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of EMNLP-IJCNLP*. Hong Kong, 3730–3740.

[123] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv arXiv:1907.11692* (2019).

[124] Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-XScience: A Large-scale Dataset for Extreme Multi-document Summarization of Scientific Articles. In *Proceedings of EMNLP*. Online, 8068–8074.

[125] Wenjuan Luo, Fuzhen Zhuang, Qing He, and Zhongzhi Shi. 2013. Exploiting relevance, coverage, and novelty for query-focused multi-document summarization. *Knowledge-Based Systems* 46 (2013), 33–42.

[126] Congbo Ma, Wei Emma Zhang, Mingyu Guo, Hu Wang, and Quan Z Sheng. 2020. Multi-document Summarization via Deep Learning Techniques: A Survey. *arXiv preprint arXiv:2011.04843* (2020).

[127] PG Magdum and Sheetal Rathi. 2021. A Survey on Deep Learning-Based Automatic Text Summarization Models. In *Advances in Artificial Intelligence and Data Engineering*. Springer, 377–392.

[128] Yuning Mao, Yanru Qu, Yiqing Xie, Xiang Ren, and Jiawei Han. 2020. Multi-document summarization with maximal marginal relevance-guided reinforcement learning. In *Proceedings of EMNLP 2020*. 1737–1751.

[129] Yogesh Kumar Meena, Ashish Jain, and Dinesh Gopalani. 2014. Survey on graph and cluster based approaches in multi-document text summarization. In *ICRAIE-2014*. IEEE, 1–5.

[130] Qiaozhu Mei, Jian Guo, and Dragomir Radev. 2010. Divrank: the interplay of prestige and diversity in information networks. In *Proceedings of ACM SIGKDD* (Washington, USA). 1009–1018.

[131] Christian M. Meyer, Darina Benikova, Margot Mieskes, and Iryna Gurevych. 2016. MDSWriter: Annotation Tool for Creating High-Quality Multi-Document Summarization Corpora. In *Proceedings of ACL-2016 System Demonstrations*. Berlin, Germany, 97–102.

[132] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to WordNet: An on-line lexical database. *International journal of lexicography* 3, 4 (1990), 235–244.

[133] Michel Minoux. 1978. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization techniques*. Springer, 234–243.

[134] Mahdi Naser Moghadasi and Yu Zhuang. 2020. Sent2Vec: A New Sentence Embedding Representation With Sentimental Semantic. In *2020 IEEE International Conference on Big Data (Big Data)*. 4672–4680.

[135] Muhidin A Mohamed and Mourad Oussalah. 2015. Similarity-based query-focused multi-document summarization using crowdsourced and manually-built lexical-semantic resources. In *IEEE Trustcom/BigDataSE/ISPA*, Vol. 2. 80–87.

[136] Shweta V Mokhale and Gauri M Dhopawkar. 2019. A Study on Different Multi-Document Summarization Techniques. In *2019 Third International Conference on Inventive Systems and Control (ICISC)*. IEEE, 710–713.

[137] Diego Mollá, Christopher Jones, and Vincent Nguyen. 2020. Query Focused Multi-document Summarisation of Biomedical Texts. *Conference and Labs of the Evaluation Forum* 2696 (2020).

[138] N Moratanch and S Chitrakala. 2016. A survey on abstractive text summarization. In *2016 International Conference on Circuit, power and computing technologies (ICCPCT)*. IEEE, 1–7.

[139] N Moratanch and S Chitrakala. 2017. A survey on extractive text summarization. In *2017 international conference on computer, communication and signal processing (ICCCSP)*. IEEE, 1–6.

[140] Tatsunori Mori, Masanori Nozawa, and Yoshiaki Asada. 2005. Multi-answer-focused multi-document summarization using a question-answering engine. *ACM Transactions on Asian and Low-Resource Language Information Processing* 4 (09 2005), 305–320.

[141] Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the ACL* 9 (2021), 1475–1492.

[142] Gopal K. R. Naveen and Prema Nedungadi. 2014. Query-Based Multi-Document Summarization by Clustering of Documents. In *Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing* (Amritapuri, India) *(ICONIAAC '14)*.

[143] N Nazari and MA Mahdavi. 2019. A survey on automatic text summarization. *Journal of AI and Data Mining* 7, 1 (2019), 121–135.

[144] Jun-Ping Ng and Viktoria Abrecht. 2015. Better Summarization Evaluation with Word Embeddings for ROUGE. In *Proceedings of EMNLP*. Lisbon, 1925–1930.

[145] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *NeurIPS 2016, Barcelona, Spain*.

[146] Milda Norkute, Nadja Herger, Leszek Michalak, Andrew Mulder, and Sally Gao. 2021. Towards Explainable AI: Assessing the Usefulness and Impact of Added Explainability Features in Legal Document Summarization. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan). ACM, 1–7.

[147] You Ouyang, Wenjie Li, Sujian Li, and Qin Lu. 2011. Applying regression models to query-focused multi-document summarization. *Information Processing & Management* 47, 2 (2011), 227–237.

[148] Sandip R Pandit and MA Potey. 2013. A query specific graph based approach to multi-document text summarization: simultaneous cluster and sentence ranking. In *2013 International Conference on Machine Intelligence and Research Advancement*. IEEE, 213–217.

[149] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL* (Philadelphia, Pennsylvania). 311–318.

[150] Sun Park, Ju-Hong Lee, Chan-Min Ahn, Jun Hong, and Seok-Ju Chun. 2006. Query Based Summarization Using Non-negative Matrix Factorization. *Lecture Notes in Artificial Intelligence* 4253, 84–89.

[151] Ramakanth Pasunuru, Asli Celikyilmaz, Michel Galley, Chenyan Xiong, Yizhe Zhang, Mohit Bansal, and Jianfeng Gao. 2021. Data Augmentation for Abstractive Query-Focused Multi-Document Summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 13666–13674.

[152] Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. 2021. Efficiently Summarizing Text and Graph Encodings of Multi-Document Clusters. *Proceedings of NAACL: HLT* (2021), 4768–4779.

[153] Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A Deep Reinforced Model for Abstractive Summarization. In *ICLR*. Vancouver.

[154] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*. 1532–1543.

[155] Laura Perez-beltrachini. 2021. Multi-Document Summarization with Determinantal Point Process Attention. *Journal of Artificial Intelligence Research* 71 (2021), 371–397.

[156] Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. 2017. Learning to Score System Summaries for Better Content Selection Evaluation.. In *Proceedings of the Workshop on New Frontiers in Summarization*. Copenhagen, 74–84.

[157] Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, 392–395.

[158] Bing Qin, Ting Liu, and Sheng Li. 2005. Survey of Multi-document Summarization [J]. *Journal of Chinese Information Processing* 6 (2005), 13–20.

[159] Pavan Kartheek Rachabathuni. 2017. A survey on abstractive summarization techniques. In *2017 International Conference on Inventive Computing and Informatics (ICICI)*. IEEE, 762–765.

[160] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 140 (2020), 1–67.

[161] Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A Multi-Pass Sieve for Coreference Resolution. In *Proceedings of EMNLP*. Cambridge, 492–501.

[162] Nazreena Rahman and Bhogeswar Borah. 2015. A survey on existing extractive techniques for query-based text summarization. In *2015 International Symposium on Advanced Computing and Communication (ISACC)*. IEEE, 98–102.

[163] K Yogeswara Rao and PV Nageswara Rao. 2016. Ontology and Query-Focused Multi-Document Summarization System. *International Journal of Computational Intelligence Research* 12, 1 (2016), 1–15.

[164] Nithin Raphal, Hemanta Duwarah, and Philemon Daniel. 2018. Survey on abstractive text summarization. In *2018 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 0513–0517.

[165] Haniyeh Rashidghalam, Mina Taherkhani, and Fariborz Mahmoudi. 2016. Text summarization using concept graph and BabelNet knowledge base. In *2016 Artificial Intelligence and Robotics (IRANOPEN)*. IEEE, 115–119.

[166] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[167] Douglas Reynolds. 2008. Gaussian Mixture Models. *Encyclopedia of Biometrics* (01 2008).

[168] Cody Rioux, Sadid A. Hasan, and Yllias Chali. 2014. Fear the REAPER: A System for Automatic Multi-Document Summarization with Reinforcement Learning. In *Proceedings of EMNLP*. Doha, 681–690.

[169] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundation and Trends in Information Retrieval* 3, 4 (apr 2009), 333–389.

[170] Haggai Roitman, Guy Feigenblat, Doron Cohen, Odellia Boni, and David Konopnicki. 2020. Unsupervised dual-cascade learning with pseudo-feedback distillation for query-focused extractive summarization. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan). 2577–2584.

[171] Mike Rosner and Carl Camilleri. 2008. MultiSum: Query-Based Multi-Document Summarization. In *COLING 2008: Proceedings of the workshop Multi-source Multilingual Information Extraction and Summarization*. Manchester, 25–32.

[172] Sascha Rothe, Joshua Maynez, and Shashi Narayan. 2021. A thorough evaluation of task-specific pretraining for summarization. In *Proceedings of EMNLP*. 140–145.

[173] Naveen Saini, Sriparna Saha, Anubhav Jangra, and Pushpak Bhattacharyya. 2019. Extractive single document summarization using multi-objective optimization: Exploring self-organized differential evolution, grey wolf optimizer and water cycle algorithm. *Knowledge-Based Systems* 164 (2019), 45–67.

[174] G.M. Salton, A. Wong, and C.S.A. Yang. 1975. A Vector Space Model for Automatic Indexing. *Commun. ACM* 18 (11 1975), 613–620.

[175] Frank Schilder and Ravikumar Kondadadi. 2008. Fastsum: Fast and accurate query-based multi-document summarization. In *Proceedings of ACL:HLT*. Columbus, 205–208.

[176] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347 (2017).

[177] Satoshi Sekine and Chikashi Nobata. 2003. *A survey for multi-document summarization*. Technical Report. New York University.

[178] Lei Sha, Baobao Chang, Zhifang Sui, and Sujian Li. 2016. Reading and thinking: Re-read lstm unit for textual entailment recognition. In *Proceedings of COLING 2016*. 2870–2879.

[179] Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond Wong, and Fang Chen. 2016. A query-based summarization service from multiple news sources. In *2016 IEEE International Conference on Services Computing (SCC)*. IEEE, 42–49.

[180] Ori Shapira and Ran Levy. 2020. Massive Multi-Document Summarization of Product Reviews with Weak Supervision. (2020). http://arxiv.org/abs/2007.11348

[181] Sheetal Shimpikar and Sharvari Govilkar. 2017. A survey of text summarization techniques for Indian regional languages. *International Journal of Computer Applications* 165, 11 (2017), 29–33.

[182] Kazutoshi Shinoda and Akiko Aizawa. 2018. Query-focused scientific paper summarization with localized sentence representation. In *Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries @ SIGIR*.

[183] Asim Sohail, Uzair Aslam, Hafiz Ilyas Tariq, and Manoj Jayabalan. 2020. Methodologies and techniques for text summarization: a survey. *Journal of Critical Reviews* 7, 11 (2020), 781–785.

[184] Andy Su, Difei Su, John M. Mulvey, and H. Vincent Poor. 2023. Optimizing Multidocument Summarization by Blending Reinforcement Learning Policies. *IEEE Transactions on Artificial Intelligence* 4, 3 (2023).

[185] Dan Su, Yan Xu, Genta Indra Winata, Peng Xu, Hyeondey Kim, Zihan Liu, and Pascale Fung. 2019. Generalizing question answering system with pre-trained language model fine-tuning. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. Hong Kong, 203–211.

[186] Dan Su, Yan Xu, Tiezheng Yu, Farhad Bin Siddique, Elham Barezi, and Pascale Fung. 2020. CAiRE-COVID: A Question Answering and Query-focused Multi-Document Summarization System for COVID-19 Scholarly Information Management. In *Proceedings of EMNLP 2020*. Online.

[187] Rui Sun, Zhenchao Wang, Yafeng Ren, and Dong-Hong Ji. 2016. Query-Biased Multi-document Abstractive Summarization via Submodular Maximization Using Event Guidance. In *Web-Age Information Management*. Springer, 310–322.

[188] Zhongkai Sun, Prathusha Sarma, William Sethares, and Yingyu Liang. 2020. Learning relationships between text, audio, and video via deep canonical correlation for multimodal language analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8992–8999.

[189] Sunaina and Sowmya Kamath S. 2016. Query-Oriented Unsupervised Multi-Document Summarization on Big Data. In *Proceedings of International Conference on Computing Communication and Networking Technologies* (Dallas, USA).

[190] Sheetal A Takale, Prakash J Kulkarni, and Sahil K Shah. 2016. An intelligent web search using multi-document summarization. *International Journal of Information Retrieval Research (IJIRR)* 6, 2 (2016), 41–65.

[191] Amol Tandel, Brijesh Modi, Priyasha Gupta, Shreya Wagle, and Sujata Khedkar. 2016. Multi-document text summarization-a survey. In *International Conference on Data Mining and Advanced Computing (SAPIENCE)*. IEEE, Ernakulam, 331–334.

[192] Oguzhan Tas and Farzad Kiyani. 2007. A survey automatic text summarization. *PressAcademia Procedia* 5, 1 (2007), 205–213.

[193] Stephen Tratz and Eduard H Hovy. 2008. Summarization Evaluation Using Transformed Basic Elements.. In *Proceedings of TAC 2008*. NIST, 10.

[194] Mohammadreza Valizadeh and Pavel Brazdil. 2015. Exploring actor–object relationships for query-focused multi-document summarization. *Soft Computing* 19, 11 (2015), 3109–3121.

[195] Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the blanc: Human-free quality estimation of document summaries. *arXiv preprint arXiv:2002.09836* (2020).

[196] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE CVPR*. Boston, 4566–4575.

[197] Byron C. Wallace, Sayantan Saha, Frank Soboczenski, and Iain J. Marshall. 2021. Generating (Factual?) Narrative Summaries of RCTs: Experiments with Neural Multi-Document Summarization. *AMIA Annual Symposium proceedings* (2021), 605–614.

[198] Haonan Wang, Yang Gao, Yu Bai, Mirella Lapata, and Heyan Huang. 2021. Exploring explainable selection to control abstractive summarization. In *AAAI Conference on Artificial Intelligence*, Vol. 35. 13933–13941.

[199] Kexiang Wang, Baobao Chang, and Zhifang Sui. 2020. A spectral method for unsupervised multi-document summarization. In *Proceedings of EMNLP 2020*. 435–445.

[200] Kexiang Wang, Tianyu Liu, Baobao Chang, and Zhifang Sui. 2020. An Anchor-Based Automatic Evaluation Metric for Document Summarization. In *Proceedings of COLING*. Barcelona (Online), 5696–5701.

[201] Rui Wang, Shijing Si, Guoyin Wang, Lei Zhang, Lawrence Carin, and Ricardo Henao. 2020. Integrating task specific information into pretrained language models for low resource fine tuning. In *Findings of the EMNLP*. 3181–3186.

[202] Furu Wei, Yanxiang He, Wenjie Li, and Qin Lu. 2008. A query-sensitive graph-based sentence ranking algorithm for query-oriented multi-document summarization. In *2008 International Symposiums on Information Processing*. IEEE, Moscow, Russia, 9–13.

[203] Furu Wei, Wenjie Li, Qin Lu, and Yanxiang He. 2008. A cluster-sensitive graph model for query-oriented multi-document summarization. In *European Conference on Information Retrieval* (Glasgow, UK). Springer, 446–453.

[204] Gary Weiss. 2005. *Data Mining and Knowledge Discovery Handbook.* Springer, 1189–1201.

[205] Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2021. PRIMER: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization. *Proceedings of ACL* (2021), 5245—-5263.

[206] Wenpu Xing and Ali Ghorbani. 2004. Weighted pagerank algorithm. In *Proceedings of Second Annual Conference on Communication Networks and Services Research*. IEEE, Fredericton, 305–314.

[207] Shufeng Xiong and Donghong Ji. 2016. Query-focused multi-document summarization using hypergraph-based ranking. *Information Processing & Management* 52, 4 (2016), 670–681.

[208] Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of ACL*. Online, 1355–1362.

[209] Yumo Xu and Mirella Lapata. 2020. Coarse-to-fine query focused multi-document summarization. In *Proceedings of EMNLP*. 3632–3645.

[210] Yumo Xu and Mirella Lapata. 2022. Document Summarization with Latent Queries. *Transactions of ACL* 10 (2022), 623–638.

[211] Pranjali Avinash Yadav-Deshmukh and R Ambekar. 2014. Survey on Multi-Document Summarization in Disaster Management based on Ontology. *International Journal of Science and Research (IJSR) ISSN (Online)* 3, 10 (2014), 2319–7064.

[212] Hiroyuki Yamauchi. 1980. Processing of Syntax and Semantics of Natural Language by Predicate Logic of Predicate Logic. In *Proceedings of COLING 1980*.

[213] Guangbing Yang. 2014. A novel contextual topic model for query-focused multi-document summarization. In *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*. IEEE, Limassol, 576–583.

[214] Guangbing Yang, Dunwei Wen, Erkki Sutinen, et al. 2013. A contextual query expansion based multi-document summarizer for smart learning. In *2013 International Conference on Signal-Image Technology & Internet-Based Systems*. IEEE, Kyoto, 1010–1016.

[215] Jen-Yuan Yeh, Hao-Ren Ke, and Wei-Pang Yang. 2006. Query-focused multidocument summarization based on hybrid relevance analysis and surface feature salience. In *Proceedings of the 6th WSEAS international conference on simulation, modelling and optimization, SMO* (Lisbon), Vol. 6. 464–469.

[216] Wenpeng Yin, Lifu Huang, Yulong Pei, et al. 2012. Relationlistwise for query-focused multi-document summarization. In *Proceedings of COLING 2012*. Mumbai, 2961–2976.

[217] Wenpeng Yin, Yulong Pei, Fan Zhang, and Lian'en Huang. 2012. Query-Focused Multi-Document Summarization Based on Query-Sensitive Feature Space. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management* (Maui, USA). 1652–1656.

[218] Seunghyun Yoon, Seokhyun Byun, and Kyomin Jung. 2018. Multimodal speech emotion recognition using audio and text. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 112–118.

[219] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of ICML*. Vienna, Austria, 11328–11339.

[220] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *ICLR 2020, Addis Ababa, Ethiopia*.

[221] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11, 3 (2020), 1–41.

[222] Jinming Zhao, Ming Liu, Longxiang Gao, Yuan Jin, Lan Du, He Zhao, He Zhang, and Gholamreza Haffari. 2020. SummPip: Unsupervised Multi-Document Summarization with Sentence Graph Compression. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and*

Development in Information Retrieval. 1949–1952.

[223] Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. In *Proceedings of EMNLP-IJCNLP*. Hong Kong, 563–578.

[224] Hai-Tao Zheng, Ji-Min Guo, Yong Jiang, and Shu-Tao Xia. 2016. Query-Focused Multi-document Summarization Based on Concept Importance. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, Cham, 443–453.

[225] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive Summarization as Text Matching. In *Proceedings of ACL*. Online, 6197–6208.

[226] Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for Effective Neural Extractive Summarization: What Works and What's Next. In *Proceedings of the ACL*. Florence, 1049–1058.

[227] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization. In *Proceedings of NAACL*.

[228] Sheng-hua Zhong, Yan Liu, Bin Li, and Jing Long. 2015. Query-oriented unsupervised multi-document summarization via deep learning model. *Expert systems with applications* 42, 21 (2015), 8146–8155.

[229] Hao Zhou, Weidong Ren, Gongshen Liu, Bo Su, and Wei Lu. 2021. Entity-Aware Abstractive Multi-Document Summarization. In *Findings of the ACL-IJCNLP 2021*. 351–362.

[230] Liang Zhou, Miruna Ticrea, and Eduard Hovy. 2004. Multi-Document Biography Summarization. In *Proceedings of EMNLP 2014*. Barcelona, 434–441.

[231] Markus Zopf. 2018. Auto-hMDS: Automatic Construction of a Large Heterogeneous Multilingual Multi-Document Summarization Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, 3228–3233.

[232] Yanyan Zou, Xingxing Zhang, Wei Lu, Furu Wei, and Ming Zhou. 2020. Pre-training for Abstractive Document Summarization by Reinstating Source Text. In *Proceedings of EMNLP*. Online, 3646–3660.

[233] Guus Zoutendijk. 1960. *Methods of feasible directions: a study in linear and non-linear programming*. Elsevier Pub. Co., Amsterdam, New York.

[234] Rolf A Zwaan, Mark C Langston, and Arthur C Graesser. 1995. The construction of situation models in narrative comprehension: An event-indexing model. *Psychological science* 6, 5 (1995), 292–297.

NUMERICAL RESULTS

Table 5, 6 and 7 presents numerical results of the scores obtained by the 9 methods on DUC 2005-07, TAC 2008-10, TD-QFS and QMSUM respectively.

Table 5. DUC 2005, 2006 and 2007

Dataset	Metric	VB15	L21	O11	K21	H06	C17	CL12	XJ16	L22
DUC 2005	ROUGE_1_F	0.22165	0.27770	0.27678	0.24827	0.23698	0.25620	0.22974	0.26413	**0.35781**
	ROUGE_2_F	0.02488	0.03932	0.04285	0.02749	0.02642	0.03188	0.01932	0.03628	**0.07366**
	ROUGE_4_F	0.00130	0.00365	**0.00418**	0.00176	0.00233	0.00241	0.00074	0.00249	0.00345
	ROUGE_L_F	0.10675	0.12552	0.13158	0.11967	0.11202	0.12195	0.10959	0.12424	**0.16452**
	ROUGE_W_F	0.03303	0.03872	0.04030	0.03624	0.03476	0.03748	0.03346	0.05443	**0.17569**
	ROUGE_S4_F	0.02991	0.04221	**0.04650**	0.03421	0.03169	0.03506	0.02679	0.03592	0.04015
	ROUGE_SU4_F	0.06215	0.08183	**0.08523**	0.07021	0.06624	0.07124	0.06096	0.07226	0.07922
	BERT_F	0.52452	0.57753	0.55770	**0.82111**	0.54288	0.56762	0.53519	0.61899	0.77402
	BLEU	0.02237	0.05012	**0.05198**	0.00031	0.03490	0.02508	0.02392	0.02623	0.00000
	CHRFPP	0.39856	**0.48880**	0.48814	0.00057	0.44502	0.41626	0.43397	0.32226	0.00076
	S3_PYR	0.24885	0.34913	**0.35396**	0.26210	0.24789	0.24837	0.18890	0.22837	-0.05225
	S3_RESP	0.37870	0.43857	**0.44398**	0.37961	0.37631	0.37750	0.33914	0.36929	0.22870
	METEOR	0.13212	0.17461	0.16414	**0.19830**	0.14517	0.14311	0.14106	0.13854	0.01435
	CIDER	0.12176	0.12001	0.11738	0.11623	0.11174	0.11869	0.11496	0.12237	**0.15449**
DUC 2006	ROUGE_1_F	0.26386	0.29223	0.29510	0.27339	0.26180	0.28163	0.25240	0.28988	**0.39036**
	ROUGE_2_F	0.03529	0.04446	0.04945	0.03189	0.03369	0.03987	0.02902	0.04570	**0.09614**
	ROUGE_4_F	0.00238	0.00329	0.00541	0.00185	0.00279	0.00304	0.00174	0.00339	**0.00628**
	ROUGE_L_F	0.12085	0.12914	0.13537	0.12338	0.11931	0.12626	0.11483	0.13178	**0.17956**
	ROUGE_W_F	0.03682	0.03943	0.04147	0.03720	0.03649	0.03832	0.03499	0.05800	**0.15264**
	ROUGE_S4_F	0.03922	0.04619	0.05119	0.03682	0.03749	0.04079	0.03356	0.04235	**0.05199**
	ROUGE_SU4_F	0.07702	0.08758	0.09222	0.07662	0.07523	0.07951	0.07040	0.08200	**0.09495**
	BERT_F	0.54285	0.57941	0.55760	**0.82528**	0.55351	0.56850	0.54169	0.62507	0.77516
	BLEU	0.03077	0.04848	**0.05353**	0.00038	0.03802	0.03025	0.02973	0.02870	0.00000
	CHRFPP	0.44173	0.49441	**0.49896**	0.00057	0.45835	0.44579	0.44984	0.33495	0.00077
	S3_PYR	0.31073	0.36611	**0.37084**	0.29462	0.28196	0.28829	0.23608	0.25828	-0.05236
	S3_RESP	0.41492	0.44772	**0.45665**	0.40264	0.39728	0.39996	0.36613	0.38762	0.22804
	METEOR	0.15332	0.17761	0.17039	**0.21119**	0.15662	0.15497	0.15146	0.14756	0.01235
	CIDER	0.11862	0.11524	0.11735	0.11651	0.11274	0.11693	0.11366	0.12147	**0.15618**
DUC 2007	ROUGE_1_F	0.28581	0.30559	0.31532	0.26444	0.29501	0.29928	0.26180	0.30356	**0.39696**
	ROUGE_2_F	0.04468	0.05682	0.06300	0.03360	0.05028	0.05295	0.03617	0.05561	**0.10469**
	ROUGE_4_F	0.00505	0.00814	**0.00897**	0.00233	0.00630	0.00616	0.00323	0.00603	0.00818
	ROUGE_L_F	0.12918	0.13877	0.14461	0.11289	0.13207	0.13494	0.11871	0.13781	**0.18841**
	ROUGE_W_F	0.03894	0.04269	0.04452	0.03509	0.04086	0.04178	0.03656	0.06101	**0.17785**
	ROUGE_S4_F	0.04672	0.05584	**0.06227**	0.03272	0.04999	0.04956	0.03665	0.04912	0.05967
	ROUGE_SU4_F	0.08698	0.09787	**0.10485**	0.07172	0.09121	0.09063	0.07458	0.09005	0.10313
	BERT_F	0.55085	0.58820	0.57803	**0.82268**	0.57568	0.58312	0.55358	0.63424	0.77064
	BLEU	0.04568	0.06409	**0.06816**	0.00039	0.05800	0.04362	0.04156	0.03970	0.00000
	CHRFPP	0.44593	0.49849	**0.50641**	0.00053	0.48691	0.45466	0.46339	0.34319	0.00067
	S3_PYR	0.32186	0.39857	**0.43421**	0.30721	0.37018	0.31453	0.28263	0.29319	-0.06231
	S3_RESP	0.42485	0.47221	**0.49751**	0.41081	0.45265	0.41798	0.39743	0.41112	0.22237
	METEOR	0.15359	0.18218	0.18204	**0.21259**	0.17492	0.16703	0.15914	0.15395	0.01319
	CIDER	0.11632	0.11585	0.11610	0.10863	0.11015	0.11596	0.10938	0.11607	**0.13608**

Table 6. TAC 2008, 2009 and 2010

Dataset	Metric	VB15	L21	O11	K21	H06	C17	CL12	XJ16	L22
TAC 2008	ROUGE_1_F	0.21153	0.21418	0.21698	0.19890	0.20608	0.21285	0.19681	0.22192	**0.30893**
	ROUGE_2_F	0.03375	0.04472	0.05067	0.02992	0.03807	0.04139	0.03141	0.04776	**0.10578**
	ROUGE_4_F	0.00492	0.00723	0.00972	0.00432	0.00527	0.00598	0.00444	0.00669	**0.01088**
	ROUGE_L_F	0.11172	0.11384	0.12122	0.09904	0.10880	0.11278	0.10271	0.11828	**0.17060**
	ROUGE_W_F	0.05040	0.05499	0.05837	0.04748	0.05223	0.05361	0.04924	0.06905	**0.16125**
	ROUGE_S4_F	0.03047	0.03958	0.04477	0.02824	0.03506	0.03578	0.02973	0.03649	**0.04754**
	ROUGE_SU4_F	0.06119	0.06905	0.07388	0.05705	0.06391	0.06501	0.05795	0.06611	**0.07976**
	BERT_F	0.53697	0.57924	0.56155	**0.83312**	0.56558	0.57241	0.55163	0.63083	0.78774
	BLEU	0.03727	0.04120	**0.04555**	0.00032	0.03522	0.03311	0.03100	0.02723	0.00001
	CHRFPP	0.46279	0.51755	**0.51818**	0.00060	0.50522	0.47859	0.49439	0.35707	0.00076
	S3_PYR	0.40155	0.55910	**0.56963**	0.43622	0.50689	0.44383	0.45144	0.41298	-0.03394
	S3_RESP	0.47928	0.58549	**0.59992**	0.48104	0.53870	0.49545	0.50167	0.48922	0.23843
	METEOR	0.16363	0.20002	0.19506	**0.26503**	0.19295	0.18814	0.18333	0.17599	0.03187
	CIDER	0.12772	0.12756	0.12872	0.12269	0.12256	0.12764	0.12083	0.12943	**0.15596**
TAC 2009	ROUGE_1_F	0.21740	0.21903	0.22547	0.20411	0.20919	0.21821	0.20480	0.22749	**0.31240**
	ROUGE_2_F	0.03453	0.04795	0.05716	0.02969	0.04112	0.04454	0.03516	0.04854	**0.09417**
	ROUGE_4_F	0.00421	0.00818	**0.01141**	0.00348	0.00637	0.00671	0.00555	0.00704	0.01009
	ROUGE_L_F	0.11643	0.11948	0.12848	0.10267	0.11105	0.11795	0.10686	0.12190	**0.16834**
	ROUGE_W_F	0.05235	0.05748	0.06184	0.04863	0.05321	0.05534	0.05093	0.07040	**0.15642**
	ROUGE_S4_F	0.03312	0.04321	**0.05077**	0.02896	0.03748	0.03848	0.03384	0.03949	0.04902
	ROUGE_SU4_F	0.06440	0.07287	0.08031	0.05850	0.06645	0.06806	0.06268	0.06966	**0.08243**
	BERT_F	0.53635	0.57905	0.56419	**0.83066**	0.56385	0.57162	0.55484	0.63093	0.78757
	BLEU	0.03413	0.04260	**0.04957**	0.00030	0.03723	0.03385	0.03357	0.02820	0.00001
	CHRFPP	0.46065	0.50842	**0.51904**	0.00059	0.50225	0.47829	0.49594	0.35536	0.00067
	S3_PYR	0.39894	0.55273	**0.59388**	0.45066	0.51326	0.45784	0.46503	0.42120	-0.02612
	S3_RESP	0.47572	0.57973	**0.61975**	0.49631	0.55238	0.50561	0.51382	0.49740	0.24408
	METEOR	0.16181	0.19795	0.19901	**0.26478**	0.19194	0.19025	0.18856	0.17650	0.03143
	CIDER	0.12596	0.12823	0.13383	0.11978	0.12272	0.12631	0.12181	0.12667	**0.13438**
TAC 2010	ROUGE_1_F	0.17478	0.20654	0.20592	0.19420	0.19725	0.20159	0.19447	0.21022	**0.29841**
	ROUGE_2_F	0.02168	0.04026	0.04541	0.02981	0.03484	0.03755	0.03032	0.04089	**0.08389**
	ROUGE_4_F	0.00147	0.00508	0.00670	0.00353	0.00522	0.00489	0.00412	0.00470	**0.00679**
	ROUGE_L_F	0.10147	0.11120	0.11472	0.10167	0.10568	0.10844	0.10119	0.11329	**0.15712**
	ROUGE_W_F	0.04018	0.05318	0.05505	0.04822	0.05055	0.05187	0.04831	0.06466	**0.15365**
	ROUGE_S4_F	0.02023	0.03651	0.03969	0.02877	0.03213	0.03223	0.02908	0.03233	**0.03986**
	ROUGE_SU4_F	0.04664	0.06521	0.06777	0.05671	0.05998	0.06038	0.05697	0.06078	**0.07216**
	BERT_F	0.50294	0.57710	0.54971	**0.83201**	0.56046	0.56878	0.55169	0.62264	0.78459
	BLEU	0.02361	0.03600	**0.03821**	0.00028	0.03474	0.02711	0.03062	0.02335	0.00001
	CHRFPP	0.29412	0.50349	**0.49883**	0.00062	0.49497	0.40863	0.49101	0.32625	0.00068
	S3_PYR	0.15402	0.51564	**0.51729**	0.42491	0.47480	0.42770	0.43048	0.35678	-0.01971
	S3_RESP	0.33783	0.55012	**0.56087**	0.48252	0.52158	0.48485	0.48718	0.45571	0.24987
	METEOR	0.09640	0.19143	0.18296	**0.25339**	0.18473	0.18211	0.18126	0.16041	0.03269
	CIDER	0.14108	0.12700	0.12621	0.12510	0.12218	0.12661	0.12135	0.12866	**0.13770**

Table 7. TD-QFS and QMSUM

Dataset	Metric	VB15	L21	O11	K21	H06	C17	CL12	XJ16	L22
TD-QFS	ROUGE_1_F	0.17323	**0.34257**	0.27424	0.32600	0.26440	0.27273	0.27725	0.27122	0.24086
	ROUGE_2_F	0.01507	**0.08977**	0.05207	0.07139	0.02943	0.05020	0.02764	0.04833	0.05296
	ROUGE_4_F	0.00025	**0.03635**	0.01208	0.01920	0.00557	0.00828	0.00354	0.01100	0.00000
	ROUGE_L_F	0.08974	**0.15863**	0.12965	0.14759	0.11346	0.12892	0.11845	0.12820	0.13286
	ROUGE_W_F	0.02568	0.05331	0.04241	0.04810	0.03654	0.04525	0.03668	0.05466	**0.13987**
	ROUGE_S4_F	0.01908	**0.08571**	0.05381	0.07008	0.03433	0.03964	0.03355	0.04495	0.01811
	ROUGE_SU4_F	0.04508	**0.12893**	0.09086	0.11310	0.07305	0.07789	0.07462	0.08116	0.04248
	BERT_F	0.48065	0.60781	0.54051	**0.82519**	0.57802	0.59291	0.55193	0.62449	0.78734
	BLEU	0.00670	**0.14416**	0.06136	0.10133	0.05212	0.05465	0.03326	0.05719	0.00140
	CHRFPP	0.21541	**0.54148**	0.48310	0.06058	0.47380	0.39623	0.46281	0.32966	0.07044
	S3_PYR	0.07011	**0.53007**	0.35949	0.50726	0.34656	0.31986	0.26437	0.29315	-0.02579
	S3_RESP	0.28299	**0.60341**	0.45004	0.56104	0.42651	0.42348	0.37286	0.42045	0.24627
	METEOR	0.06448	0.20446	0.15625	**0.28681**	0.16617	0.15318	0.14869	0.15011	0.02394
	CIDER	**0.14806**	0.12681	0.12181	0.12225	0.12331	0.12506	0.11848	0.12874	0.14045
QMSUM	ROUGE_1_F	0.06928	0.14408	0.15265	0.15014	0.14280	0.14711	0.15715	0.13875	**0.15514**
	ROUGE_2_F	0.00698	0.01572	0.01756	0.02617	0.01655	0.01812	0.01868	0.01891	**0.03074**
	ROUGE_4_F	0.00022	0.00107	0.00129	**0.00310**	0.00108	0.00113	0.00103	0.00128	0.00118
	ROUGE_L_F	0.04493	0.08284	0.08613	0.09174	0.08290	0.08514	0.08734	0.08414	**0.11309**
	ROUGE_W_F	0.01570	0.04241	0.04399	0.04835	0.04261	0.04435	0.04472	0.05012	**0.10462**
	ROUGE_S4_F	0.00739	0.01807	0.01954	**0.02744**	0.01848	0.01901	0.02064	0.01870	0.01932
	ROUGE_SU4_F	0.01782	0.03940	0.04205	**0.04821**	0.03952	0.03946	0.04375	0.03846	0.03844
	BERT_F	0.70071	0.70349	0.69178	**0.79824**	0.70252	0.71092	0.71835	0.72839	0.78366
	BLEU	0.00491	0.00893	0.00847	**0.01242**	0.00882	0.00864	0.00972	0.00846	0.00595
	CHRFPP	0.05148	0.05425	0.05530	**0.06031**	0.05469	0.05390	0.05222	0.05355	0.04660
	S3_PYR	0.00603	0.27039	0.27207	**0.37150**	0.29065	0.27123	0.32826	0.21062	-0.06457
	S3_RESP	0.25472	0.37698	0.38020	**0.44271**	0.39147	0.37859	0.40981	0.35352	0.21878
	METEOR	0.05738	0.17130	0.17708	**0.18976**	0.16982	0.17056	0.18607	0.13845	0.01770
	CIDER	0.10317	0.10735	0.10995	**0.11963**	0.10844	0.10679	0.10321	0.10623	0.09183