

Using Query Expansion in Manifold Ranking for Query-Oriented Multi-Document Summarization

Quanye Jia
Beihang University
jiaquanye@buaa.edu.cn

Rui Liu
Beihang University
lr@buaa.edu.cn

Jiaying Lin
Beihang University
jiaying.lin@buaa.edu.cn

Abstract

Manifold ranking has been successfully applied in query-oriented multi-document summarization. It not only makes use of the relationships among the sentences, but also the relationships between the given query and the sentences. However, the information of original query is often insufficient. So we present a query expansion method, which is combined in the manifold ranking to resolve this problem. Our method not only utilizes the information of the query term itself and the knowledge base WordNet to expand it by synonyms, but also uses the information of the document set itself to expand the query in various ways (mean expansion, variance expansion and TextRank expansion). Compared with the previous query expansion methods, our method combines multiple query expansion methods to better represent query information, and at the same time, it makes a useful attempt on manifold ranking. In addition, we use the degree of word overlap and the proximity between words to calculate the similarity between sentences. We performed experiments on the datasets of DUC 2006 and DUC2007, and the evaluation results show that the proposed query expansion method can significantly improve the system performance and make our system comparable to the state-of-the-art systems.

1 Introduction

Query-focused multi-document summarization is to create a summary from a set of documents that answers the information requirements expressed in the query. Compared to generic summarization, query-focused summarization requires the summary biased to a specific query besides the general requirement for a summary. In contrast to the task of question answering (QA) that mainly focuses on simple factoid questions and results in precise answers such as person, location or date, etc., in the case of query-focused summarization, the queries are mostly real-world complex questions and the information provided by the queries is insufficient. So how to understand the query and expand the query has a certain impact on the quality of the query-oriented summary.

Currently most query oriented abstract systems are based on general summarization systems. By incorporating features that are related to the given query (e.g., the relevance of a sentence to the query), a generic summarization system can be adapted to a query-focused one. Due to the limit of information that a query can express, some systems also expand the query using some external resources such as WordNet, by which the synonyms of the query words can be obtained as expansion words (Zhou et al., 2005). However, such approaches to query expansion are restricted in query itself and external resources, because they cannot be applied to words not in WordNet such as named entities which frequently occur in queries, and much context related information cannot be captured by only synonyms. Lin (Lin et al., 2009) uses graph-based sentence ranking and sentence-to-word relationships to implement query expansion, which can improve the effect of graph ranking. However, the paper uses a single query expansion method and only adopts TextRank instead of manifold ranking as graph ranking algorithm. So we improved the query expansion method based on the above problems.

The main contributions of this work are:

©2021 China National Conference on Computational Linguistics
Published under Creative Commons Attribution 4.0 International License

- We used multiple query expansion methods to expand the query, including the query itself (using Wordnet synonyms) and the expanded query based on the document set (mean expansion, variance expansion and TextRank expansion).
- We improved the calculation of sentence similarity matrix in the manifold ranking process, using sentence TF-ISF similarity, word overlap similarity, and sentence proximity information.

Our experiments on the dataset DUC 2006 and DUC 2007 show that the summarization results with query expansion are much better than that without query expansion, achieving the state-of-the-art performance.

2 Related Work

Graph-based ranking algorithms have been successfully used in text summarization. (G. and Radev, 2004) proposed LexRank for generic text summarization. They construct a connected similarity graph where nodes represent sentences and edges represent cosine similarities between sentences. A random walk is applied on the graph until converging to a stationary distribution, by which the sentences can be ranked. Topic-sensitive LexRank(Otterbacher et al., 2005) has been applied to the task of query-focused summarization where the relevance of a sentence to the query is taken into account when performing random walk. (Xiong and Ji, 2016) proposed a hypergraph based vertex-reinforced random walk framework for multi-document summarization. (Wan et al., 2007) applied a manifold ranking algorithm to query-focused summarization which can simultaneously make full use of both the relationships among all the sentences in the documents and the relationships between the given query and the sentences.

Regarding manifold ranking algorithms, subsequent researchers have made a lot of improvements. (Tan et al., 2015a) introduced the matrix factorization model on the basis of Xiaojun Wan's model, and constructed the sentence relational matrix by using the cosine value of the decomposed matrix. (Chali et al., 2011) used the syntactic and shallow semantic kernels to calculate the correlation between sentences based on the model of Xiaojun Wan. On the basis of Xiaojun Wan's model, (Lin et al., 2019) improves the relationship matrix by mixing the cosine similarity of decomposed matrix and TF-ISF cosine similarity in a certain proportion. At the same time, this paper uses lifelong learning topic model to enhance the co-occurrence effect of matrix factorization words. Finally, this paper adds statistical features, mix various scoring methods together, and then extract summaries with the strategy of de-redundancy. (Cai and Li, 2012) not only considers the internal relevance propagation within the sentence set (or within the theme cluster set), but also considers the mutual reinforcement between the ranking of the sentence set and the ranking of the cluster set. These models mainly focus on the calculation of sentence similarity and sentence ranking, but pay little attention to queries.

Query understanding is a research hotspot in the field of search engines. (Abdelali et al., 2007) proposed the use of a corpus to expand the original query. (Jian et al., 2009) proposed a Wikipedia-based user query intent understanding. (Pal et al., 2014) considered the distribution of expansions in documents, statistical associations with query terms, and semantic associations.(Wang and Kong, 2011) and (Pinto et al., 2009) combined query expansion with word meaning elimination to enhance the validity of the query. To the best of our knowledge, few people have done the research on the effect of query understanding and query expansion on manifold ranking. Therefore, based on the above research, we use multiple query expansion methods to improve the effect of model based on manifold ranking.

3 Method

In this section, this paper will introduce the model architecture, as Fig.1 shows.

For the preprocessing step, this paper performed sentence segmentation, word segmentation, word stemming and TF-ISF calculation for each sentence in the documents and the query. This paper utilized titles and narratives as queries. When computing the sentence similarities, stop words, emails, digital characters, punctuation marks and single characters were removed, and only the content words (i.e., noun, verb, adjective and adverb) were used after the morphological analysis based on treetagger.

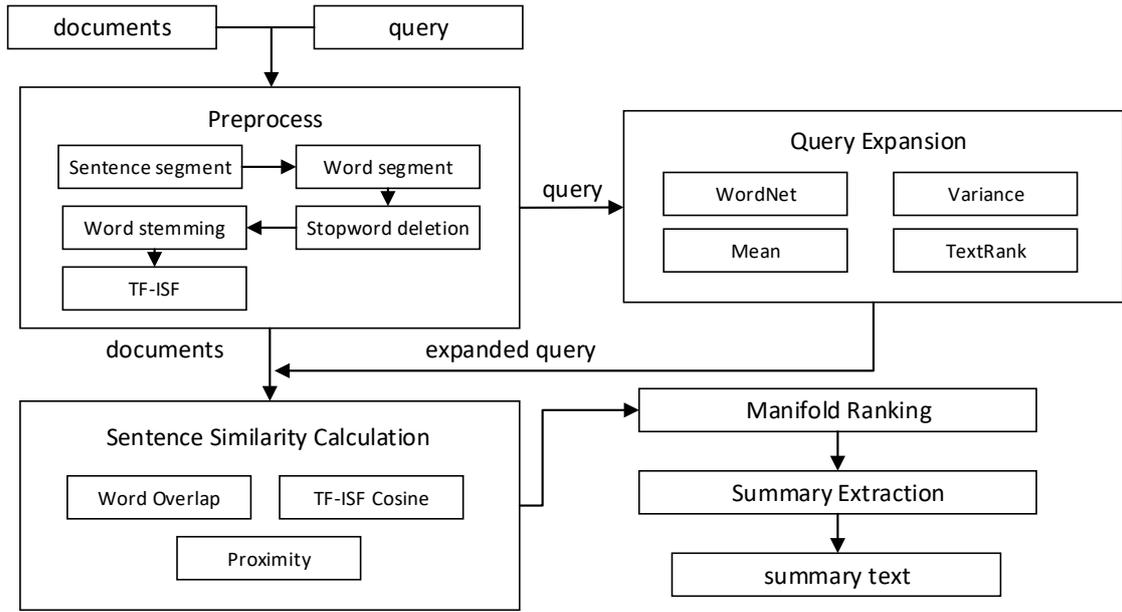


Figure 1: Model architecture.

For the extraction process of automatic text summarization, this paper selects the method used by (Wan et al., 2007).

Manifold Ranking, query expansion and sentence similarity calculation and will be introduced in detail in following sections.

3.1 Manifold Ranking

Manifold ranking (Zhou et al., 2004a) (Zhou et al., 2004b) is a universal graph ranking algorithm and it is initially used to rank data points along their underlying manifold structure. The prior assumption of manifold-ranking is: (1) sentences on the same structure are likely to have the same ranking scores; (2) sentences similar to the query sentence have higher ranking scores. The process of the manifold ranking algorithm can be understood as follows: construct a similarity matrix W between all sentences, initialize a vector $y = [y_0, y_1, \dots, y_n]^T$, in which $y_0 = 1$ because x_0 is the query sentence and $y_i = 0 (1 \leq i \leq n)$ for all the sentences in the documents. Then the ranking algorithm transmits the scores of query sentences to the scores of sentences similar to query points through the similarity between sentences, and then the scores of each sentence are propagated to the scores of other similar sentences, iterating many times until the scores of each sentence remain unchanged. The algorithm for manifold ranking is as Formula 1 shows:

$$L(f) = \alpha_{mr} * \sum_{i,j=1}^N W_{ij} \left| \frac{1}{\sqrt{D_{ii}}} f_i - \frac{1}{\sqrt{D_{jj}}} f_j \right|^2 + (1 - \alpha_{mr}) * \sum_{i=1}^N |f_i - y_i|^2$$

$$\text{s.t. } D_{ii} = \text{sum}(W_{i*}), Y = [y_1 \ y_2 \ \dots \ y_N]^T = [1 \ 0 \ \dots \ 0]^T \quad (1)$$

Where $F = [f_1 \ f_2 \ \dots \ f_n]^T$ is the sentence score vector for the solution, and Y is the vector indicating the query sentence. This paper assumes that when A_i is the query sentence or the query sentence $y_i = 1$, and otherwise is 0, W_{ij} indicates the relationship between the i -th sentence and the j -th sentence. There are many ways to calculate the relationship between sentences, such as Manhattan

distance, Euler distance, inner product calculation method, cosine similarity, etc., or a combination of various methods. D_{ij} represents the sum of the elements of the i -th row in the W matrix, and α_{mr} represents the proportion of the similarity between the sentences in the manifold ranking.

3.2 Query Expansion

In the query-oriented multi-document summarization, each topic has one or two sentences as the query, but the information of these sentences is often insufficient. Therefore, the query sentence should be expanded. There are two general methods: one is to expand according to the content of documents. The other is to expand based on the query itself, which usually depends on the understanding of the query. For each topic, we extract query sentences and sentences of multiple documents to form a sentence-word matrix $A^{(0)} = [T^{(0)} \ X]^T$ of $N \times M$, and then use the above two methods to expand the query sentence, the formula for expanding query sentences is as Formula 2 shows:

$$T^{(1)} = \text{sim_word}(T^{(0)}) + \theta_m \text{mean}(X) + \theta_v \text{variance}(X) + \theta_r \text{TextRank}(A^{(0)}) \quad (2)$$

Where A is a sentence-word matrix, the word weight is calculated using the TF-ISF(as (Lin et al., 2009) use) formula, T is query sentences, X is all document sentences, N is the number of the sentences, M is the number of all various content words in the documents, $\text{sim_word}(T^{(0)})$ is the expansion of the word meaning of the query sentence, $\text{mean}(X)$ is the mean expansion of the query sentence, $\text{variance}(X)$ is the variance expansion of the query sentence, $\text{TextRank}(A^{(0)})$ is the TextRank expansion of the query sentence.

3.2.1 Query Expansion By WordNet

WordNet is a large English word sense database(Fellbaum and Miller, 1998). It not only explains the meaning of words, but also links the meaning of words together. In WordNet, each of synset(semantic) represents a basic lexical concept. A word can have more than one synonym, and a synonym can correspond to more than one word. Synonymous phrases are connected by semantic relations such as hypernymy-hyponymy, synonym-antonym, meronymy-holonyms and implication, forming a network structure with synonymous phrases as nodes and semantic relations as edges. Therefore, semantic similarity can be calculated by the length of the edge between two semantic nodes.

Semantic similarity calculation is based on (Rada et al., 1989), which mainly uses the path length between concepts. The shorter the path between two nodes is, the higher the semantic similarity between the two concepts is. In this paper, the similarity is calculated using the following formula 3:

$$\text{sim}(\text{syn}_1, \text{syn}_2) = \frac{a}{a + d} \quad (3)$$

Where $\text{syn}_1, \text{syn}_2$ refers to the semantic node of a word, d represents the number of upper edges of the shortest path between the semantic nodes syn_1 and syn_2 in the semantic graph, and a represents an adjustable parameter.

For most applications, there is no data with word meaning labels, so algorithms are needed to provide us with similarities between words rather than between meanings or concepts. For any algorithm based on a semantic dictionary, according to Resnik (Resnik, 1995), we can approximate the similarity by using the maximum similarity between two word semantic items. Therefore, based on word meaning similarity, we can define word similarity as following formula 4:

$$\text{sim}(w_1, w_2) = \max_{\substack{\text{syn}_1 \in \text{senses}(w_1) \\ \text{syn}_2 \in \text{senses}(w_2)}} \text{sim}(\text{syn}_1, \text{syn}_2) \quad (4)$$

Where w_1, w_2 represent the target word, $\text{syn}_1, \text{syn}_2$ refer to the sense of the word. For a sentence-word matrix $A^{(0)}$ of $N \times M$, the similarity of all words in the document can be obtained first, and then

the word similarity matrix of $M \times M$ can be obtained. According to the matrix, each word of the query sentence can be expanded. The expanded formula is as Formula 5 shows.

$$t_i^{(1)} = \max_j(t_j^{(0)} \cdot \text{sim}(w_j, w_i)) \quad (5)$$

$t_i^{(0)}$ represents the i -th word feature before expansion, $t_i^{(1)}$ represents the expanded i -th word feature, and w_i represents the i -th word item, then $T^{(1)} = \text{sim_word}(T^{(0)}) = \begin{bmatrix} t_1^{(1)} & t_2^{(1)} & \dots & t_M^{(1)} \end{bmatrix}$ is the query sentence after the word meaning expansion. By expanding the meaning of the query sentence, it can make the query sentence reflect the features of the similar words in the original sentence, and solve the problem that the feature of the query sentence is not salient because the original query sentence has too few words.

3.2.2 Word Similarity Filtering

According to the above word similarity calculation method, we can extract all words from the dataset to calculate the word similarity matrix. However, most of the similarities between words have no practical significance, so they need to be filtered. There are two kinds of filtering methods in this paper: vertical filtering and horizontal filtering.

Vertical filtering is based on the path length between words. For example, computer and football are obviously not similar, but because they can be connected by superordinate entities, even if the shortest path between them is very long, they will still get smaller values after similarity calculation. This degree of similarity has little significance for feature calculation, so smaller values can be discarded. In this way, not only computational meaningful features can be expanded, but also sparse matrix storage can be used to reduce the waste of storage space. Therefore, this paper will use vertical filtering parameter as L to filter the similarities which the shortest path is greater than L .

Horizontal filtering is based on the number of similarities between a word and other words. When a word is similar to most words, it shows that the feature of the word itself is not salient, so it needs to be filtered. For example, the verbs "use" and "do" are similar to almost all verbs. If these verbs are expanded, noise will be introduced into the textual features of the query sentences, which will not highlight the topic feature of the query sentences. Therefore, this paper will use horizontal filtering parameter as C to retain the words which are similar to no more than C other words.

3.2.3 Query expansion By Mean and Variance

Due to the large amount of information contained in the text, the average TF-ISF value and the variance TF-ISF value of each word in the document can be obtained to expand the query sentence, as shown in the following Formula 6 and 7 show.

$$\text{mean}(X) = [\bar{X}_{*1} \quad \dots \quad \bar{X}_{*M}] = \left[\frac{1}{N-1} \sum_{i=1}^{N-1} X_{i1} \quad \dots \quad \frac{1}{N-1} \sum_{i=1}^{N-1} X_{iM} \right] \quad (6)$$

$$\text{variance}(X) = \left[\frac{1}{N-2} \sum_{i=1}^{N-1} (X_{i1} - \bar{X}_{*1})^2 \quad \frac{1}{N-2} \sum_{i=1}^{N-1} (X_{i2} - \bar{X}_{*2})^2 \quad \dots \quad \frac{1}{N-2} \sum_{i=1}^{N-1} (X_{iM} - \bar{X}_{*M})^2 \right] \quad (7)$$

N represents the number of all sentences, X_{ij} represents the i -th document sentence and the j -th word item feature, \bar{X}_{*j} represents the average of the j -th word item feature in the documents.

3.2.4 Query expansion By TextRank

In this section, we select from the document set both informative and query relevant words based on TextRank (section 3.2 in (Lin et al., 2009)) results, add them into the original query and use the updated query to perform manifold ranking again. Our query expansion approach goes as follows:

1. Normalize X by $S = D^{-1}X$ to make the sum of each row equal to 1, where D is the diagonal matrix with (i, i)-element equal to the sum of the i th row of X .

2. Calculate vector y by $y = S^T p^*$, where p^* is the vector of sentence ranking scores derived in the last step in the TextRank algorithm described in Lin et al. (Lin et al., 2009), and y represents the word scores.

3. Rank all the words based on their scores in y and select the top c words as query expansions. c is a parameter representing the number of expansion words, which is set in the experiments.

4. Add the top c words into the query sentence T , the formula is as follows:

$$\text{TextRank}(A^{(0)}) = [t'_1, t'_2, \dots, t'_n] \quad (8)$$

$$t'_i = \begin{cases} 1 & \text{when the words corresponding to } t \text{ are in the top } c \text{ words} \\ 0 & \text{when the words corresponding to } t \text{ are not in the top } c \text{ words} \end{cases} \quad (9)$$

Where t'_i represents the i -th word feature after expansion. This algorithm is to make use of both the sentence importance and sentence-to-word relationships to select the expansion words. By this step, salient words occurring in the important sentences are more likely to be selected, and because the higher ranked sentences are biased towards the query, the words selected in this way are also biased towards the query.

3.3 Sentence Similarity Calculation

The W matrix is used to measure the similarity between two sentences, which is critical in manifold ranking. If the calculation method of W is closer to the similarity between sentences, the quality of the summary is higher. This paper mainly uses TF-ISF cosine similarity, word overlap and proximity similarity to calculate. The TF-ISF cosine similarity and word overlap use the word frequency characteristics of the sentence itself. Word overlap only considers the number of words co-occurring in two sentences, regardless of the occurrence of words in other sentences, which is beneficial to increase the number of different words in the summary. The adjacency similarity mainly considers the relationship between the relative positions of two sentences, so that the meaning of the sentence in the context is reflected. The three are organically combined, as shown in the following Formula 10:

$$W_{ij} = \alpha_A \cdot \cos(A_{i*}^{(1)}, A_{j*}^{(1)}) + \alpha_{overlap} \cdot SS_{ij} + \alpha_{peer} \cdot P_{ij} \quad (10)$$

Where $\cos(A_{i*}^{(1)}, A_{j*}^{(1)})$ represents the cosine similarity of the TF-ISF of the sentence $A_{i*}^{(1)}$ and $A_{j*}^{(1)}$. SS_{ij} indicates the degree of word overlap of the sentence $A_{i*}^{(0)}$ and $A_{j*}^{(0)}$, that is, the ratio of the number of words appearing together in two sentences to the minimum word length of two sentences.

For the position information of the sentence, this paper mainly considers the context of the sentence. P_{ij} is used to represent the adjacency matrix of the sentence. One topic has multiple documents, and one document has multiple sentences. These sentences constitute the context and the sentences are in a specific document. There is a specific context, each sentence must have a certain relationship with the adjacent sentence, so the adjacency matrix can be used to quantify the relationship between the sentences. The closer the sentences are, the higher the similarity score is. The representation is as shown in the following Formula 11:

$$P_{ij} = \begin{cases} 0.1^{|i-j|}, & i \text{ and } j \text{ are in the same document} \\ 0, & i \text{ and } j \text{ are not in the same document} \end{cases} \quad (11)$$

In addition, when solving the cosine similarity matrix of TF-ISF between sentences, this paper uses the expanded query sentence as the query sentence. When calculating the sentence word overlap matrix, the original query sentence is used as the query sentence.

Dataset	Extension Method	Rouge-1	Rouge-2	Rouge-W	Rouge-SU4
DUC2006	ori	0.40331	0.08809	0.13960	0.14637
	ori+MEAN	0.40695	0.09057	0.14099	0.14827
	ori+VAR	0.40604	0.09013	0.14060	0.14835
	ori+SIM_WORD	0.41019	0.08785	0.14084	0.14691
	ori+TextRank	0.41658	0.09174	0.14274	0.15102
	ori+SIM_WORD+MEAN	0.41129	0.08895	0.14078	0.14762
	ori+SIM_WORD+VAR	0.41112	0.08879	0.14113	0.14762
	ori+MEAN+VAR				
	+SIM_WORD	0.41060	0.08934	0.14090	0.14723
	ori+MEAN+VAR +SIM_WORD+TextRank	0.41674	0.09202	0.14279	0.15071
DUC2007	ori	0.42010	0.10372	0.14559	0.16051
	ori+MEAN	0.42612	0.10638	0.14773	0.16329
	ori+VAR	0.42149	0.10444	0.14616	0.16102
	ori+SIM_WORD	0.42565	0.10195	0.14564	0.15991
	ori+TextRank	0.43557	0.11174	0.15088	0.16668
	ori+SIM_WORD+MEAN	0.42907	0.10447	0.14706	0.16163
	ori+SIM_WORD+VAR	0.42699	0.10184	0.14604	0.16016
	ori+MEAN+VAR				
	+SIM_WORD	0.43042	0.10574	0.14768	0.16295
	ori+MEAN+VAR +SIM_WORD+TextRank	0.43982	0.11185	0.15159	0.16870

Table 1: ROUGE Evaluation Results of Expanded Query Sentences and Original Query Sentences in DUC2006 and DUC2007.

4 Experimental Setup

4.1 Datasets and Evaluation

The summary data sets used in this experiment are DUC2006(Dang, 2006) and DUC2007(Dang, 2007). DUC 2006 and DUC 2007 are query-oriented multi-document summary datasets, each data set contains multiple topics, each topic consists of multiple related documents, and each topic provides a title and a narrative as a query. For each topic, we take the first 250 words as the model summary from the results and compare it with the expert summary by using ROUGE(Lin and Hovy, 2003) toolkit. We report F1 scores of ROUGE-1, ROUGE-2, ROUGE-W and ROUGE-SU4 metrics.

4.2 Parameter settings

The parameters of the proposed methods are determined according to the overall effect of the model on DUC data set. The parameter a is 1, L is 4, C is 5000, the mean parameter θ_m is 1, the variance parameter θ_v is 1, r_t is 0.4, the α_{mr} is 1, the cosine similarity parameter α_A is 0.9, the word coverage parameter $\alpha_{overlap}$ is 0.1, and the adjacency similarity parameter α_{peer} is 0.4. ω as a de-redundancy parameter, takes 8 (refer to (Wan et al., 2007)). In the TextRank expansion, the c is 100 and d is 0.6 (refer to (Lin et al., 2009)).

5 Results

5.1 Comparison with query expansion methods

In this experiment, our aim is to examine the efficiency of the combination of these query expansion methods. The results are reported in table 1.

From the report, we can figure out that the performance of the method using expansion of query is

better than that using the original query. In other words, the results of ori+MEAN+VAR+TextRank are better than one that combines a subset of query expansion methods.

Dataset	System	Rouge-1	Rouge-2	Rouge-W	Rouge-SU4
DUC2006	Random	0.35352	0.05381	0.12135	0.111
	Lead	0.3396	0.05412	0.11748	0.10961
	MV-CNN	0.3865	0.0791	-	0.1409
	QODE	0.4015	0.0928	-	0.1479
	AttSum	0.409	0.094	-	-
	VAEs-A	0.396	0.089	-	0.143
	C-Attention	0.393	0.087	-	0.141
	MultiMR	0.40306	0.08508	0.13997	-
	SingleMR	0.39934	0.07502	0.13445	0.13404
	JMFMR	0.41244	0.0887	-	0.14585
	JTMMR	0.40014	0.08160	0.13652	0.14062
	JLTMMR	0.4043	0.08126	0.13774	0.14032
	JLTMMR+SF	0.41045	0.08926	0.14214	0.14798
	RDRP_AP	0.39615	0.08975	-	0.13905
	TextRank	0.39056	0.0871	0.13536	0.14299
	QE-T+TextRank	0.40115	0.08985	0.13804	0.14619
QE-WMVT+Mani	0.41674	0.09202	0.14279	0.15071	
DUC2007	Random	0.36896	0.06654	0.125	0.12312
	Lead	0.35461	0.06639	0.1219	0.11926
	Kmeans+NMF	0.37076	0.07194	0.12786	0.12814
	MV-CNN	0.4092	0.0911	-	0.1534
	QODE	0.4295	0.1163	-	0.1685
	AttSum	0.4392	0.1155	-	-
	VAEs-A	0.421	0.11	-	0.164
	C-Attention	0.423	0.107	-	0.161
	MultiMR	0.42041	0.10302	0.14595	-
	SingleMR	0.41422	0.09052	0.13984	0.14589
	JTMMR	0.42717	0.10181	0.1458	0.15761
	JLTMMR	0.43349	0.10375	0.14786	0.16002
	JLTMMR+SF	0.43734	0.10439	0.1496	0.1625
	RDRP_AP	0.43775	0.11563	-	0.16904
	HERF	0.42444	0.11211	-	0.16307
	TextRank	0.4033	0.10196	0.14063	0.15463
QE-T+TextRank	0.41554	0.10597	0.14379	0.16074	
QE-WMVT+Mani	0.43982	0.11185	0.15159	0.16870	

Table 2: Rouge results of multi-document summarization on DUC2006 and DUC2007.

5.2 Comparison with related methods

In this experiment, we will compare the model performance with the other existing well-known methods. We compare QE-WMVT-Mani(Manifold ranking with word similarity, mean, variance and TextRank query expansion) with other methods: 1)Random(For each topic, randomly extracts a certain number of sentences as a summary); 2)Lead(Under each topic, select a certain number of sentences in the document of the most recent time as a summary); 3)MV-CNN(Zhang et al., 2016); 4)AttSum(Cao et al., 2016); 5)VAEs-A(Li et al., 2017b); 6)QODE(Liu et al., 2015); 7)C-Attention(Li et al., 2017a); 8)MultiMR(Wan and Xiao, 2009); 9)JMFMR(Tan et al., 2015b); 10)JTMMR(Lin et al., 2019); 11)JLTMMR(Lin et al., 2019); 12)JLTMMR + SF(Lin et al., 2019); 13)RDRP_AP(Cai and Li, 2012); 14)HERF(Xiong and Ji,

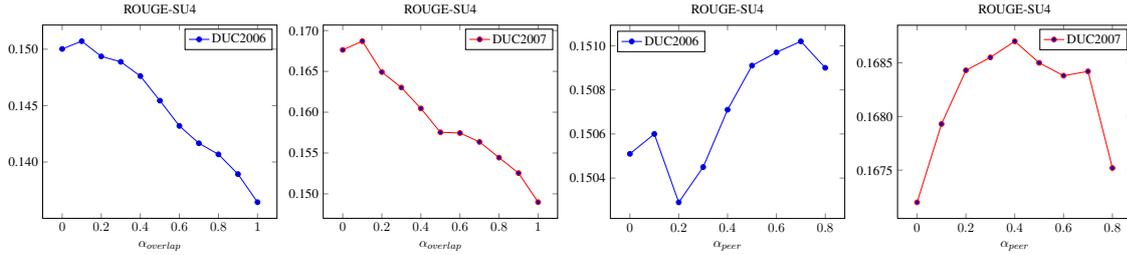


Figure 2: ROUGE-SU4 scores vs. $\alpha_{overlap}$ ($\alpha_{overlap} + \alpha_A = 1$) and α_{peer} .

2016); 15)TextRank(Mihalcea and Tarau, 2004); 16)QE-T+TextRank(Lin et al., 2009).

For MV-CNN, QODE, AttSum, VAEs-A, C-Attention, MultiMR, JMFMR, JTMMR, JLTMMR, JLTMMR+SF, RDRP_AP and HERF models, this paper chooses results from their corresponding papers. For the QE-T+TextRank method, because the original paper does not give the F value, this paper re-experiments it, and the preprocessing process is the same as the original paper. r_t is set to 0.4, and other parameters are the same as the original paper.

Table 2 shows the ROUGE scores of related methods. The bolded results highlight the best results in the set of experiments. From the results, we have the following observations:

1. Our proposed method (QE-WMVT+Mani) outperforms other related methods generally. Random and Lead sentences show the poorest ROUGE scores.

2. The effect of deep learning method is not very prominent in the query-oriented multi-document summarization, only slightly prominent in the score of ROUGE-2, and significantly lower in the score of ROUGE-1 than our proposed method. This shows that the method of deep learning needs to be improved.

3. Manifold ranking models based on query sentence expansion (QE-WMVT + Mani) have improved compared with the manifold ranking model MultiMR, JMFMR, JTMMR, JLTMMR, JLTMMR+SF and RDRP_AP. This shows that the manifold ranking based on the expansion of the query sentence is generally better than the manifold ranking based on the original query sentence.

4. TextRank with TextRank expansion(QE-T+TextRank) can improve the effect of TextRank, but the overall performance is not as good as our method. This is because the TextRank method of the original query sentence is significantly worse than manifold ranking of the original query sentence (see result in section 5.1) and the query expansion method has limited ability to improve the model.

5.3 Influence of Sentence Similarity Parameter Tuning

In our method, $\alpha_{overlap}$ is used to tune the trade-off between the TF-ISF and word overlap feature. α_{peer} is utilized to add the context feature. We carry out experiments with different $\alpha_{overlap}$ and α_{peer} to see their influence.

Fig.2 presents the ROUGE-SU4 evaluation results of our method on DUC2006 and DUC2007 respectively for different values of $\alpha_{overlap}$ and α_{peer} . In general, with an increase in $\alpha_{overlap}$ and α_{peer} , the performance first increases, reaches its peak value and then is degraded. For DUC2006 and DUC2007, a value of $\alpha_{overlap}$ around 0.1 can attain the best ROUGE-SU4 result. This indicates that TF-ISF is a critical feature in sentence similarity, but word overlap feature can improve the effect of the model. For the DUC2006 dataset, a value of α_{peer} around 0.7 can attain the best ROUGE-SU4 result. For DUC2007, $\alpha_{peer} = 0.4$ produces peak values of ROUGE scores. Although the best value of α_{peer} differ for DUC2006 and DUC2007, they can slightly improve the model results. This indicates that there are positional connections between sentences.

5.4 Query Relevance Performance

We also perform the qualitative analysis to exam the query relevance performance of our model. We randomly choose some queries in the test datasets and calculate the relevance scores of sentences from our model. We then extract the top ranked sentences and check whether they are able to meet the query

<p><i>Query:</i> What has been the argument in favor of a line item veto? How has it been used? How have US courts, especially the Supreme Court, ruled on its constitutionality?</p>
<p><i>Our model without query expansion Summary:</i> Supreme Court to Rule on Line Item Veto. Court rules Line Item Veto unconstitutional. Court strikes down president’s Line Item Veto power. Supreme Court Rules Line Item Veto Law Unconstitutional. Clinton Vetoes Congress Rejection of His Line Item Veto.</p>
<p><i>QE-WMVT+Mani Summary:</i> Supreme Court to Rule on Line Item Veto. In a 6-3 decision the high court ruled the Line Item Veto Act violated the constitution’s separation of powers between Congress which approves legislation and the president who either signs it into law or vetoes it. Court strikes down president’s Line Item Veto power. Clinton is the first president to have line item veto authority. In a 6-3 ruling the court said the 1996 Line Item Veto Act unconstitutionally allows the president by altering a bill after its passage to create a law that was not voted on by either house of Congress.</p>

Table 3: Sentences recognized to focus on the query.

need. Examples for query and model summaries are shown in Table 3. We also give the sentences from our model without query expansion for comparison.

With manual inspection, we find that most query-focused sentences in our model with query expansion can answer the query to a large extent. All these aspects are mentioned in reference summaries. The sentences from our model without query expansion, however, are usually short and simply repeat the key words in the query. The advantage of query expansion is apparent in query relevance ranking.

Different from single document summary, multi-document summary has more information, and the central idea of each document under each topic is different. Therefore, the amount of valid information, such as ROUGE, is more suitable for evaluating the effect of multi-document summarization. Although our model is based on extraction method, it is simple and efficient, and the amount of valid information can be comparable to the state-of-the-art models.

6 Conclusion

In this paper, we propose a query-oriented multi-document summarization model based on query expansion and manifold ranking. We use WordNet, mean, variance, TextRank to expand query. We also use the cosine values of TF-ISF, word overlap values and the relative position of sentences to calculate the relation between sentences. We applied our model to DUC2006 and DUC2007, and found that our model is better than one that combines a subset of query expansion methods and achieves competitive performance with the state-of-the-art results.

References

- Ahmed Abdelali, Jim Cowie, and Hamdy S. Soliman. 2007. Improving query precision using semantic expansion. *Information Processing and Management*, 43(3):705–716.
- X. Cai and W. Li. 2012. Mutually reinforced manifold-ranking based relevance propagation model for query-focused multi-document summarization. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(5):1597–1607.
- Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2016. Attsun: Joint learning of focusing and summarization with neural attention.
- Yllias Chali, Sadid A Hasan, and Kaisar Imam. 2011. Using syntactic and shallow semantic kernels to improve multi-modality manifold-ranking for topic-focused multi-document summarization. *Ijcnlp*.

- Hoa Trang Dang. 2006. Overview of duc 2006. In *In Document Understanding Conference*.
- Hoa Trang Dang. 2007. Overview of duc 2007. In *In Document Understanding Conference*.
- Christiane Fellbaum and George Miller. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- Erkan G. and D Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence*, 22(1):457–479.
- Jian, Wang, Gang, Lochovsky, Fred, T. Jian, Chen, and Zheng. 2009. Understanding user’s query intent with wikipedia. *Www Madrid! Track Search*, pages 471–480.
- Piji Li, Wai Lam, Lidong Bing, Weiwei Guo, and Hang Li. 2017a. Cascaded attention based unsupervised information distillation for compressive summarization. pages 2081–2090, 01.
- Piji Li, Zihao Wang, Wai Lam, Zhaochun Ren, and Lidong Bing. 2017b. Salience estimation via variational auto-encoders for multi-document summarization. In *AAAI 2017*, pages 3497–3503.
- Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1, pages 71–78.
- Zhao Lin, Lide Wu, and Xuanjing Huang. 2009. Using query expansion in graph-based approach for query-focused multi-document summarization. *Information Processing and Management*, 45(1):35–41.
- Jiaying Lin, Rui Liu, and Quanye Jia. 2019. Joint lifelong topic model and manifold ranking for document summarization.
- Yan Liu, Shenghua Zhong, and Wenjie Li. 2015. Query-oriented multi-document summarization via unsupervised deep learning. *Expert Systems with Applications*, 2, 06.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proc Conference on Empirical Methods in Natural Language Processing*.
- Jahna Otterbacher, Gunes Erkan, and Dragomir R Radev. 2005. Using random walks for question-focused sentence retrieval. pages 915–922.
- Dipasree Pal, Mandar Mitra, and Kalyankumar Datta. 2014. Improving query expansion using wordnet. *Journal of the Association for Information Science and Technology*, 65(12):2469–2478.
- Francisco Joao Pinto, Antonio Farina Martinez, and Carme Fernandez Perez-Sanjulian. 2009. *Joining automatic query expansion based on thesaurus and word sense disambiguation using WordNet*.
- Roy Rada, Hamed Mili, Ellen Bicknell, and Maria Blettner. 1989. Development and application of a metric on semantic nets. 19(1):17–30.
- Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy.
- Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2015a. Joint matrix factorization and manifold-ranking for topic-focused multi-document summarization. In *International Acm Sigir Conference on Research and Development in Information Retrieval*.
- Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2015b. Joint matrix factorization and manifold-ranking for topic-focused multi-document summarization. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 987–990. ACM.
- Xiaojun Wan and Jianguo Xiao. 2009. Graph-based multi-modality learning for topic-focused multi-document summarization. pages 1586–1591, 01.
- Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Manifold-ranking based topic-focused multi-document summarization. In *International Joint Conference on Artificial Intelligence*.
- Ruiqin Wang and Fansheng Kong. 2011. Semantic query expansion based on unsupervised word sense disambiguation. *Journal of the China Society for Scientific and Technical Information*.
- Shufeng Xiong and Donghong Ji. 2016. Query-focused multi-document summarization using hypergraph-based ranking. *Information Processing and Management*, 52(4):670–681.

- Yong Zhang, Meng Joo Er, Rui Zhao, and Mahardhika Pratama. 2016. Multiview convolutional neural networks for multidocument extractive summarization. *IEEE transactions on cybernetics*, 47:3230–3242.
- Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2004a. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16(3).
- Dengyong Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet, and Bernhard Schölkopf. 2004b. Ranking on data manifolds. In *Advances in neural information processing systems*, pages 169–176.
- Liang Zhou, Chin-Yew Lin, and Eduard Hovy. 2005. A be-based multi-document summarizer with query interpretation. In *Proceedings of Document Understanding Conference, Vancouver, BC, Canada*. Citeseer.

JCL 2021