

Density-Based Graph Model Summarization: Attaining better Performance and Efficiency

Mohammadreza Valizadeh^{*1}, Pavel Brazdil^{*+}

^{*}LIAAD laboratory, INESC TEC, University of Porto, Portugal

⁺ FEP, University of Porto, Portugal

Abstract. Several algorithms based on PageRank algorithm have been proposed to rank the document sentences in the multi-document summarization field and LexRank and T-LexRank algorithms are well known examples. In literature different concepts such as weighted inter-cluster edge, cluster-sensitive graph model and document-sensitive graph model have been proposed to improve LexRank and T-LexRank algorithms (e.g. DsR-G, DsR-Q) for multi-document summarization. In this paper, a density-based graph model for multi-document summarization is proposed by adding the concept of density to LexRank and T-LexRank algorithms. The resulting generic multi-document summarization systems, DensGS and DensGSD were evaluated on DUC 2004 while the query-based variants, DensQS, DensQSD were evaluated on DUC 2006, DUC 2007 and TAC 2010 task A. ROUGE measure was used in the evaluation. Experimental results show that density concept improves LexRank and T-LexRank algorithms and outperforms previous graph-based models (DsR-G and DsR-Q) in generic and query-based multi-document summarization tasks. Furthermore, the comparison of the number of iterations indicates that the density-based algorithm is faster than the other algorithms based on PageRank.

Keyword: Unsupervised summarization, multi-document summarization, graph-based approach to summarization, performance of summarization, document density.

1 Introduction

Document summarization generates a short text for single or multiple documents. This summary should be effective, efficient, informative and irreducible. It means that summary should cover important concepts of the original document or documents and should not add unnecessary details. Extractive and abstractive methods are two main approaches to summarize documents automatically. Extractive methods concatenate the important sentences to the summary. Abstractive methods exploit language processing and rephrasing of sentences.

Graph-based models are widely used in extractive multi-document summarization systems. They represent the sentences of a given document set as a graph in which nodes represent sentences and

¹ Corresponding author: Mohammadreza Valizadeh, LIAAD – INESC Tec, Campus da FEUP, Rua Dr. Roberto Frias, 378,4200 - 465 Porto, Portugal.

Email: mohammadreza.valizadeh@inescporto.pt

edges the similarities between sentences. The cosine measure is normally used to represent the similarity. Computing the salience of the sentences is the next step. Graph-based algorithms which extract global information from the graph can be used. The final step involves identifying the sentences with high saliency and then concatenating them into the summary.

One of the most popular graph-based algorithms for computing the salience of sentences is LexRank [1] which is based on PageRank algorithm [2, 3]. There are some improved algorithms in graph-based multi-document summarization that follow this line and include the concepts of *documents* and a *query*. Document-sensitive graph-based model for multi-document summarization [4] and cluster-sensitive graph-based model for query-oriented multi-document summarization [5] are some examples of this direction.

When a graph is generated for a set of documents, some information will be omitted because the graph is only able to show the similarities between sentences. It cannot take into account which sentence is related to which document, how many sentences there are, or how many inter- or intra-links exist in each document.

Previous work shows that additional information such as the concept of document and inter- and intra-links can improve the summaries. Researchers have added some of these concepts to the PageRank algorithm to improve the quality of summaries [4, 6]. However, the effect of the document density has been neglected by the previous graph-based approaches in the summarization field. Therefore, we study its usefulness in the process of generating a better summary for a given document set.

When a document is *dense*, its sentences are close / similar to each other. Our work is based on an assumption that the higher the density of a document, the higher the salience of its sentences.

One method to compute the density is by using the inverse of radius. When the radius of a document is small, its density is high. Furthermore, when a sentence is *central, close to the document centroid*, it has more effect on the related document density.

In [6] the researchers introduced a centroid-based summarization. Obviously the centroid value is different from density. We show that document density leads to better results than the centroid value.

Experimental results show that a summarization system that exploits the concept of density outperforms previous graph-based models in generic and query-based multi-document summarization tasks. Furthermore, density-based algorithms need fewer iterations to converge.

The rest of this paper is organized as follows. Section 2, discusses the previous work on graph-based multi-document summarization. In section 3, we review LexRank and T-LexRank algorithms. In section 4, the proposed idea is introduced and density-based concept is explained. In section 5, iteration reduction of sentence selection algorithms is explored. The experiments and evaluations are presented in section 6. Finally, section 7 concludes the paper.

2 Related Work

Many methods have been proposed for the multi-document summarization and most of them are extractive. Sentence ranking algorithm based on salience has the main role in these methods. There are many sentence ranking algorithms described in previous studies.

Graph-based sentence ranking algorithm is one of the well-known methods. It is based on PageRank [2, 3] and HITS [7] algorithms which are the two famous algorithms in this area. In the recent years, researchers have applied these algorithms to multi-document summarization.

Erkan and Radev [1] applied PageRank algorithm to a weighted undirected graph. They took the sentences as the *vertices* and cosine similarity between sentences as the weights of the links respectively. The system was referred as LexRank.

Other researchers used LexRank as a basis and added more information to it. In another words, they improved LexRank system by adding some additional concepts and exploiting them. Divya Padmanabhan et al. [8] proposed that the graph of the documents can have two types of links. They introduced inter- and intra-links. When two sentences that join two vertices belong to one document, their link is an intra-link and otherwise it is an inter-link. They postulated that inter-links are more important than intra-links and they reported that using this concept can improve the graph and the summary leading to better results based on ROUGE measure [9].

Furu Wei et al. [4, 5] added document and query concepts to the LexRank and showed that this way better summaries are obtained. To improve LexRank, they applied: (1) The centroid weight of words [6] to the algorithm in generic summarization task; (2) Similarity of two documents in query-based multi-document summarization task. This lead to improved results.

Wan et al. [10] proposed a topic-sensitive graph-based model, T-LexRank, that was used for a query-based multi-document summarization. They used two graphs to show inter- and intra-links in query-oriented multi-document summarization.

Zhao et al. [11] used query expansion in graph-based approach for query-focused multi-document summarization. They used topic-sensitive LexRank [12] (i.e. T-LexRank) twice, once for selecting terms that expand the query and second time for selecting the sentences for the summary. In other words, all sentences of the collection are ranked by T-LexRank algorithm and all terms in the collection are given a specific weight based on the sentence rank. Afterwards, the query is extended by those terms with a higher weight. Finally, T-LexRank algorithm applies the extended query to the collection and ranks the sentences again. Ouyang et al. [13] proposed a progressive sentence selection strategy for document summarization. They intended to ensure the coverage of the summary by an intuitive idea. They considered the uncovered concepts only when the saliency of the sentences was being measured.

This paper applies the concept of density to graph-based multi-document summarization. The method described in this paper improves the summaries informativeness and the system run-time as well. Run-time has been neglected in most previous work on multi-document summarization. The method proposed reduces the number of iterations of the algorithm of sentence selection (e.g. LexRank and T-LexRank). In the next sections, the algorithms and the methods are explained in more details.

3 LexRank & T-LexRank-Based Summarization Review

The bases of the above mentioned algorithms and our new extensions are LexRank (Erkan et al. [1]) and T-LexRank. Therefore, in this section we present a review of these two algorithms.

The framework of LexRank or T-LexRank can be explained as follows.

- Documents are preprocessed to separate sentences and represent them using the bag-of-words representation. Further preprocessing is applied.
- An undirected graph based on the preprocessed sentences is generated.
- The weighted neighbor matrix (M) based on cosine similarity for the generated graph is computed.
- LexRank is used for generic summarization and T-LexRank is used for query-based summarization to compute the rank of nodes (i.e. sentences) of the graph.
- The sentences with highest rank are selected to create the summary. The summary should have a specific length and sentence duplication is not allowed. Many summarization systems simply reject sentences if its similarity with previous selected sentences is equal or greater than 0.7.

Fig.1 shows the algorithm used by LexRank to compute the rank of sentences belonging to a graph. As we can note the algorithm does this in an iterative manner. It includes a cycle (on line 5) that re-computes the rank of all sentences (line 7) until a stopping condition is verified (line 10).

The rank $R(S_i)$ of each sentence S_i is calculated on line 7. As this is the basis of the method, the corresponding equation is referred to as equation (1).

The term N represents the total number of the nodes in a graph, d is a damping factor which is typically 0.85, and $N(S)$ denotes the set of neighboring sentence vertices of S and $sim(S_i, S_j)$ is the similarity between document S_i and S_j .

```

1  Input: M, N number of nodes of graph, threshold, damping factor d
2  Output: An array R of LexRank scores
3   $R_0 = \frac{1}{N} \mathbf{1}$  (1 denotes a  $N \times 1$  vector of all 1's)
4  t=0
5  repeat
6      t=t+1
7      for  $S_i=1$  to N
8          
$$R_t(S_i) = \frac{(1-d)}{N} + d * \sum_{S_j \in N(S_i)} \left( R_{t-1}(S_j) * \frac{sim(S_i, S_j)}{\sum_{S_k \in N(S_j)} sim(S_j, S_k)} \right)$$
 (1)
9      end
10      $\gamma = |R_t - R_{t-1}|$ 
11 until  $\gamma >$  threshold
12 return R

```

Fig. 1. LexRank algorithm

Otterbacher et al. [14] adapted LexRank algorithm to topic-sensitive multi-document summarization. This algorithm is known as T-LexRank and it is used as the basis for our new extension.

Fig. 2 shows the whole algorithm which computes the rank of sentences belonging to a graph.

```

1 Input: Array of sentences D; M, N number of nodes of graph, threshold, damping factor d, query Q,
Output: An array L of LexRank scores
2  $R_0 = \frac{1}{N} \mathbf{1}$ 
3  $t = 0$ 
4 repeat
5      $t = t + 1$ 
6     for  $S_i = 1$  to N
7          $R_t(S_i) = (1 - d) * \text{sim}(D(S_i), Q) + d * \sum_{S_j \in N(S_i)} \left( R_{t-1}(S_j) * \frac{\text{sim}(S_i, S_j)}{\sum_{S_k \in N(S_j)} \text{sim}(S_j, S_k)} \right)$  (2)
8     end
9      $\gamma = |R_t - R_{t-1}|$ 
10 until  $\gamma > \text{threshold}$ 
11 return R

```

Fig. 2. T-LexRank algorithm

In Fig. 2, the term $R(S_i)$ denotes the rank of the i -th sentence (S_i) that is computed on line 7. Function $\text{sim}(D(S_i), Q)$ calculates the cosine similarity between a given query Q and $D(S_i)$ representing sentence S_i of document D . If we enrich these equations so that they would use more information, they will be able to compute the ranking score more accurately. The next section discusses this in more detail.

4 Density-Based Summarization

Density of a document expresses how much its elements (i.e. sentences) are close to each other. When the sentences in a document are similar, the density of the document is high.

We hypothesize that for summarization a document with higher density is more useful. When a document has a higher density, it means that the document has a specific issue which many sentences are addressing. However, when a document has a lower density, the document usually addresses different issues. Therefore, this document and its sentences are less important in the summarization task. This paper uses density concept to improve the multi-document summarization. This algorithm is referred as *Density-Based Summarization*.

4.1 Computing Density

We define two new concepts – the *centrality of the sentence* and the *density of the document*. The centrality of the sentence R_{Xk} can be computed as the Euclidian distance between the sentence and its document centroid as follows:

$$R_{Xk} = |\vec{X}_k - \vec{X}_0| \quad (3)$$

where vector \vec{X}_0 is the centroid and vector \vec{X}_k represents a sentence.

Document density is computed as an inverse of document radius, as there is an inverse relationship between density and radius. Fig. 3 shows this.

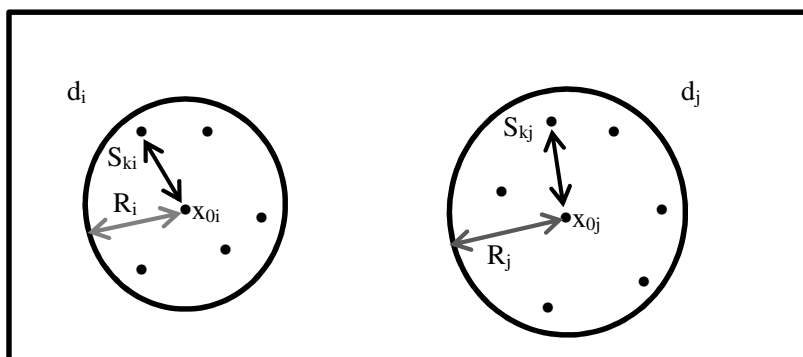


Fig.3. Example of two documents, d_i and d_j . Document d_i is more dense than d_j .

In the figure above d_i and d_j represent documents, S_{k_i} , S_{k_j} sample sentences, x_{0_i} and x_{0_j} the document centroids and R_i , R_j the document radiuses. . We hypothesize that if a sentence is closer to its document centroid (has higher centrality), or if it comes from a dense document, its rank should be higher than those of other sentences. Therefore, we have adapted the existing ranking algorithms to take this into account.

The document radius R can be calculated on the basis of the sentence centralities of each document as follows:

$$R = \left(\frac{\sum_{k=1}^N (\vec{X}_k - \vec{X}_0)^2}{N} \right)^{\frac{1}{2}} \quad (4)$$

where N is the number of sentences, vector \vec{X}_0 is the centroid and vector \vec{X}_k represents a sentence. The centroid is defined as:

$$\vec{X}_0 = \frac{\sum_{k=1}^N \vec{X}_k}{N} \quad (5)$$

where the symbols have the same meaning as before.

Radius is different from the centroid value. Radev et al. [6] proposed centroid-based summarization and introduced centroid value for each sentence. They computed the centroid value of a sentence as the sum of the centroid values of all words in the sentence. Centroid value of a

word $w_{j,s}$ is equal to $w_{j,c}$ (i.e. w_j of the centroid), if word w_j appears in sentence s , but it is 0 if it does not. Furu Wei et al. [4] used this definition and applied it in their system.

We apply the density to the basic ranking algorithms (e.g. LexRank and T-LexRank) in two steps. First, the centrality of the sentence is applied and so we obtain one algorithm. After that, the document density is added and so obtain another and more complex algorithm. The aim is to show that by adding more information to the process of ranking we can obtain improvements of the system performance.

4.2 Density & Graph-Based Ranking Algorithm for Generic Multi-Document Summarization

In LexRank (equation 1), the edges are processed without taking into account the information about documents. As was mentioned, some researchers added this type of information to this equation and improved the result.

Furu Wei et al. [4] added the *centroid value* and the fact that sentences belong to *different documents* to LexRank. The basic algorithm is the same as shown in Fig. 1, but the equation online 7 has been altered. Here we show the altered equation that forms the basis of their ranking algorithm (DsR-G).

$$R(S_i) \frac{(1-d)}{N} * (centro_{norm(d_i)}) + d * \sum_{S_j \in N(S_i)} \left(R(S_j) * \frac{sim(S_i, S_j)}{\sum_{S_k \in N(S_j)} sim(S_j, S_k)} sim_{norm(d_i, d_j)} \right) \quad (6)$$

where $centro_{norm(d_i)}$ is the *normalized centroid value* of document d_i and $sim_{norm(d_i, d_j)}$ is normalized similarity between document d_i and d_j . Furthermore, S_i is a sentence of d_i and S_j is a sentence of d_j .

One of the concepts that has not been covered in previous research is the concept of density. First, we apply the *centrality of the sentence* to the LexRank ranking algorithm to obtain a new ranking algorithm (DensGS), defined as follows:

$$R(S_i) = \frac{(1-d)}{N} + d * \sum_{S_j \in N(S_i)} \left(R(S_j) * \frac{sim(S_i, S_j)}{\sum_{S_k \in N(S_j)} sim(S_j, S_k)} * \frac{1}{1+r_i} \right) \quad (7)$$

As in previous work, $R(S_i)$ denotes the rank of sentence S_i and $N(S_j)$ represents the set of neighboring sentences of S_j . Furthermore, r_i is the normalized radius of S_i (i.e. radius value of a document normalized for that document) that is calculated as the Euclidian distance between S_i and the centroid of the document to which S_i belongs. The centroid of each document and the radius of all sentences are calculated in the initialization phase.

We hypothesize that the sentences which are closer to the document centroid are more important and so the ranking algorithm should rank them higher. Our proposed density-based algorithm changes the influence of the links by their related radius.

The ranking algorithm *DensGS* was extended further by incorporating document density. The following equation shows this ranking algorithm (*DensGSD*).

$$R(S_i) = \frac{(1-d)}{N} + d * \sum_{S_j \in N(S_i)} \left(R(S_j) * \frac{\text{sim}(S_i, S_j)^{\frac{1}{1+R_j}}}{\sum_{S_k \in N(S_j)} \text{sim}(S_j, S_k)} * \frac{1}{1+r_i} \right) \quad (8)$$

where R_j is the radius of document j .

The equation above shows that if the neighbor sentences (e.g. S_j) belong to dense documents, they should affect more the sentence (e.g. S_i) rank.

4.3 Density & Graph-Based Ranking Algorithm for Query-Based Multi-Document Summarization

Furu Wei et al. [4] extended T-LexRank by incorporating information that sentences belong to different documents. The following equation shows their algorithm of revising the rank of sentences

$$(DsR-Q): R(S_i) = (1 - d) * \text{sim}(S_i, Q) + d * \sum_{S_j \in N(S_i)} \left(R(S_j) * \frac{\text{sim}(S_i, S_j)}{\sum_{S_k \in N(S_j)} \text{sim}(S_j, S_k)} * \text{sim}_{norm}(d_i, d_j) \right). \quad (9)$$

Using the concept of centrality of the sentence in the previous equation leads to the following new ranking algorithm (DensQS):

$$R(S_i) = (1 - d) * \text{sim}(S_i, Q) + d * \sum_{S_j \in N(S_i)} \left(R(S_j) * \frac{\text{sim}(S_i, S_j)}{\sum_{S_k \in N(S_j)} \text{sim}(S_j, S_k)} * \frac{1}{1+r_i} \right). \quad (10)$$

The method above can be further extended by adding the document density. The following equation shows this ranking algorithm (*DensQSD*).

$$R(S_i) = (1 - d) * \text{sim}(S_i, Q) + d * \sum_{S_j \in N(S_i)} \left(R(S_j) * \frac{\text{sim}(S_i, S_j)^{\frac{1}{1+R_j}}}{\sum_{S_k \in N(S_j)} \text{sim}(S_j, S_k)} * \frac{1}{1+r_i} \right). \quad (11)$$

The following table shows the characteristics of all the algorithms discussed above:

Table 1. Comparison of the algorithms

	Generic	Query_based	Centrality of the sentence	Document density	Document centroid
LexRank	*				
T-LexRank		*			
DsR-G	*				*
DsR-Q		*			
DensGS	*		*		
DensQS		*	*		
DensGSD	*		*	*	
DensQSD		*	*	*	

In the following section, we examine the effect of adding density to sentence selection algorithms on the reduction of the number of iterations.

4.4 Considerations Regards Convergence

This section discusses convergence of sentence ranking algorithms.

The basis of LexRank, T-LexRank and our algorithms is PageRank. The PageRank ranking scheme is defined as:

$$R = d \cdot \vec{P} + (1 - d) \cdot M \cdot R \quad (13)$$

where R denotes the ranking vector, d is the damping factor between 0 and 1, M denotes the normalized affinity matrix of similarity graph and \vec{P} denotes the preference probability vector where each element is positive and the sum of all elements is 1.

Other algorithms (i.e. LexRank, T-LexRank and our proposed algorithms) have the same ranking scheme presented in Eq. (13) and they differ in their different use of M and \vec{P} .

PageRank algorithm is based on Markov chain and P is transition matrix and can be found by the eigenvector [15].

Wei et al. [4] stated that P is both irreducible and stochastic as well as, P is primitive because P is positive. The authors proved that the dominant eigenvector of P is unique with 1 as the eigenvalue. based on Perron's theorem [16].

Wei et al. [4] proved that P is a preference probability vector. Therefore, we should only make the matrix M column-stochastic and irreducible.

Density-based algorithms are based on an assumption that a sentence that is closer to the document center (i.e. is more central) should be ranked higher, therefore, the density affects the M matrix, as it is multiplied by $\frac{1}{1+r_i}$ and $\frac{1}{1+R_j}$ (i.e. r_i is radius of sentence i in its own document and R_j is radius of document j) to the i^{th} column.

We know that all columns in matrix M are normalized and any column sums to 1. Furthermore, if there are zero columns they are replaced with the preference vector P as in PageRank.

As we know, M is column-stochastic if the weight matrix M is column-stochastic. Therefore M is column-stochastic because each column sums to 1.

The final requirement is that M is irreducible. Since the graph that is related to M is strongly connected thus M is irreducible. Thus, we can compute vector P and the power iteration method applied to P converges to R .

5 Iteration Reduction

Run-time is a very important factor of algorithms. This has been disregarded in previous studies involving the PageRank algorithm and the follow-up summarization algorithms based on it. The previous work was normally only concerned with a measure of how good a summary is and the aim

was to improve it. This section explores the effect of density on the reduction of the number of iterations of the sentence ranking algorithms.

As we know, PageRank algorithm and its extensions (i.e. LexRank and T-LexRank) are recursive and iterative. Therefore, they are time-consuming.

Our aim is to show that the algorithms that use more information than previous methods (i.e. PageRank, LexRank and T-LexRank), require a fewer number of iterations. In addition, we show that DensGSD and DensQSD algorithms that have been extended by incorporating the concept of document density take fewer iterations when compared to previous algorithms.

We examine this idea in conjunction with the work done by Furu Wei et al. [4] that explored the notion of the centroid value and the concept that sentences belong to different documents. Experiments described in the next section confirm that the additional information (i.e. density) leads to a decrease of the number of iterations.

6 Experiments and Results

We have proposed algorithms both for generic and query-based multi-document summarization. Therefore, the experiments were set up on appropriate datasets. DUC 2004 [17] was used for generic multi-document summarization, while DUC 2006, DUC 2007 and TAC 2010 task A was used for query-based multi-document summarization.

Both DUC 2004 and DUC 2006 have 50 document sets each, DUC 2007 has 45 document sets and TAC 2010 has 46 document sets. Each document set of DUC 2004, 2006, 2007 and TAC 2010 has 10, 25, 25 and 10 documents, respectively. All the documents and queries of DUC 2006, DUC 2007 and TAC 2010 category descriptions have been pre-processed by sentence segmentation and word splitting. Words were stemmed by Porter Stemmer and stop-words were removed. TF*IDF and cosine similarity measure were used to compute the similarity of sentences for all four data sets and the similarity of sentences and queries or category descriptions with DUC 2006, DUC 2007 and TAC 2010. To avoid the link-by-chance problem that occurs when two sentences share one or two common words, we set a small threshold, 0.05, and do not consider the links which have a lower value than this threshold.

After the sentence ranking the top-ranked sentences from the original documents are selected and concatenated into the summary until the byte limitation has been reached. Following the definition of the generic summarization task, our system generates the generic summaries which are limited to 665 bytes for DUC 2004. The query-based summary for DUC 2006 and DUC 2007 is limited to 250 words and for TAC 2010 task A to 100 words. When the incoming sentence is very similar to the previously selected sentences (i.e. the sentence is redundant), it is discarded. In the experiments the similarity threshold was set to 0.7.

Automatic evaluation method, ROUGE-1.5.5 was used with parameters: `-e -n 2 -x -a -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -d` [19].

6.1 Results of Density-Based Algorithms

We compare our generic algorithms (i.e. DensGS and DensGSD) with LexRank and DsR-G algorithm proposed by Furu et al. [4] that uses the centroid value. Furthermore, we compare our query-based algorithms (i.e. DensQS and DensQSD) with T-LexRank and DsR-Q [4] algorithms that

use the document concept. The following sections show the results.

6.1.1 Results of Algorithms Based on Centrality of the sentence

This section shows the results of the algorithms based on the centrality of the sentence (i.e. equations 7 and 10). Table 2a and Table 2b show the results of evaluation of DensGS on DUC 2004 and comparisons to other systems. The subsequent three pairs of tables show the results of DensQS on three different datasets and comparisons to other systems. We have compared our system to the results of the 35 participating systems of DUC 2004, 32 participating systems of DUC 2006, 32 participating systems of DUC 2007 and with 23 participating systems of TAC 2010.

Table 2a) Model evaluation on DUC 2004 data set.

	ROUGE-2	ROUGE-SU4
LexRank	0.0853	0.1279
DsR-G	0.0872	0.1290
DensGS	0.0879	0.1317

Table 2b) Comparison with participating systems.

	ROUGE-2	ROUGE-SU4
Human	0.0860	0.1390
S65	0.0922	0.1333
S67	0.0906	0.1310
S66	0.0887	0.1308
DensGS	0.0879	0.1317
NIST Baseline	0.0529	0.1162

Table 3a) Model evaluation on DUC 2006 data set.

	ROUGE-2	ROUGE-SU4
T-LexRank	0.0856	0.1394
DsR-Q	0.0899	0.1427
DensQS	0.0907	0.1444

Table 3b) Comparison with participating systems.

	ROUGE-2	ROUGE-SU4
Human	0.1036	0.1683
S24	0.0956	0.1553
S15	0.0910	0.1473
DensQS	0.0907	0.1444
S12	0.0898	0.1476
NIST Baseline	0.0403	0.0872

Table 4a) Model evaluation on DUC 2007 data set.

	ROUGE-2	ROUGE-SU4
T-LexRank	0.1051	0.1560
DsR-Q	0.1123	0.1682
DensQS	0.1140	0.1690

Table 4b) Comparison with participating systems.

	ROUGE-2	ROUGE-SU4
Human	0.1300	0.1845
S15	0.1245	0.1771
S29	0.1203	0.1707
S4	0.1189	0.1700
DensQS	0.1140	0.1690
NIST Baseline	0.0403	0.0872

Table 5a) Model evaluation on TAC 2010 data set.

	ROUGE-2	ROUGE-SU4
T-LexRank	0.0693	0.1122
DsR-Q	0.0765	0.1190
DensQS	0.0875	0.1285

Table 5b) Comparison with participating systems.

	ROUGE-2	ROUGE-SU4
Human	0.0962	0.1381
S22	0.0957	0.1301
S23	0.0940	0.1296
DensQS	0.0875	0.1285
S18	0.0942	0.1241
S24	0.0920	0.1283
S36	0.0919	0.1227
NIST Baseline	0.0538	0.0857

As tables 2-5 show, our systems based on centrality of the sentence (DensGS) outperform the baseline systems and also ranks well among the competing systems in DUC 2004. Furthermore, it shows at least 2.6% and 0.8% improvement compared to LexRank and DsR-G algorithms, respectively. These results show that the proposed idea is useful for generic multi-document summarization.

Furthermore, our proposed query-based variant (DensQS) outperforms the baseline systems and also ranks well in DUC 2006, DUC 2007 and TAC 2010.

6.1.2 Results of Algorithms Based on Document Density and Centrality of the Sentence

This section shows results of the algorithms based on centrality of the sentence and document density together (i.e. equations 8 and 11). Therefore, we expect these algorithms will have better performance. The focus here is on DensGSD and DensQSD and their performances are compared to DensGS and DensQS on different datasets. Our system was also compared to with other participating systems and the corresponding results are shown in tables 6 to 9.

Table 6a) Model evaluation on DUC 2004 data set.

	ROUGE-2	ROUGE-SU4
DensGS	0.0879	0.1317
DensGSD	0.0923	0.1357

Table 6b) Comparison with participating systems.

	ROUGE-2	ROUGE-SU4
Human	0.0860	0.1390
DensGSD	0.0923	0.1357
S65	0.0922	0.1333
S67	0.0906	0.1310
S66	0.0887	0.1308
NIST Baseline	0.0529	0.1162

Table 7a) Model evaluation on DUC 2006 data set.

	ROUGE-2	ROUGE-SU4
DensQS	0.0907	0.1444
DensQSD	0.0930	0.1482

Table 7b) Comparison with participating systems.

	ROUGE-2	ROUGE-SU4
Human	0.1036	0.1683
S24	0.0956	0.1553
DensQSD	0.0930	0.1482
S15	0.0910	0.1473
S12	0.0898	0.1476
NIST Baseline	0.0403	0.0872

Table 8a) Model evaluation on DUC 2007 data set.

	ROUGE-2	ROUGE-SU4
DensQS	0.1140	0.1690
DensQSD	0.1210	0.1709

Table 8b) Comparison with participating systems.

	ROUGE-2	ROUGE-SU4
Human	0.1300	0.1845
S15	0.1245	0.1771
DensQSD	0.1210	0.1709
S29	0.1203	0.1707
S4	0.1189	0.1700
NIST Baseline	0.0403	0.0872

Table 9a) Model evaluation on TAC 2010 data set.

	ROUGE-2	ROUGE-SU4
DensQS	0.0875	0.1285
DensQSD	0.0923	0.1342

Table 9b) Comparison with participating systems.

	ROUGE-2	ROUGE-SU4
Human	0.0962	0.1381
DensQSD	0.0923	0.1342
S22	0.0957	0.1301
S18	0.0942	0.1241
S23	0.0940	0.1296
S24	0.0920	0.1283
S36	0.0919	0.1227
NIST Baseline	0.0538	0.0857

Tables 6-9 show the results of the algorithms that include the document density concept. As we would expect, the results exceed the values of the simpler variant that does not include this concept. It means that those algorithms that use both the centrality of sentences and document density have better performance than the algorithms that use just the first concept.

Tables 6-9 confirm this. The ranking methods DescGSD and DensQSD rank well among the participating systems (e.g. 1st on DUC 2004, 2nd on DUC 2006 etc.). Our system is not always the best, as the participating systems use other additional means, like syntactical and semantic analysis, to improve the summarization. Our work here focused on how density can improve summarization.

The results shown confirm our hypothesis that addition of more information to the algorithms

improves their performance.

6.2 Results on Iteration Reduction

Other experiments were concerned with iteration reduction. We compared our algorithms with LexRank, T-LexRank and Furu's algorithms and focused on number of iterations. The results are shown in Figures 4-7. The horizontal axis shows the document set number and the vertical axis the number of iterations required for the algorithm to reach convergence. The brackets on the right show the maximum, mean and the minimum for each algorithm.

The figures show that the addition of density to previous algorithms improves not only the measures of accuracy, but also reduces the number of iterations compared to other methods. Although DsR-G and DsR-Q that use more information than LexRank and T-LexRank also reduce the number of iterations, the reductions are larger for density based algorithms. It means that density can add more useful information to LexRank and T-LexRank when compared to centroid values and the document concept. Furthermore, the figures show that the algorithms that use both centrality of the sentence and document density reduce number of iterations more than the algorithms that only use the centrality of the sentence.

The results show the algorithms based on both centrality of the sentence and document density (e.g. DensGSD and DensQSD) are, on average, about 8 times faster than the base algorithms (i.e. LexRank and T-LexRank).

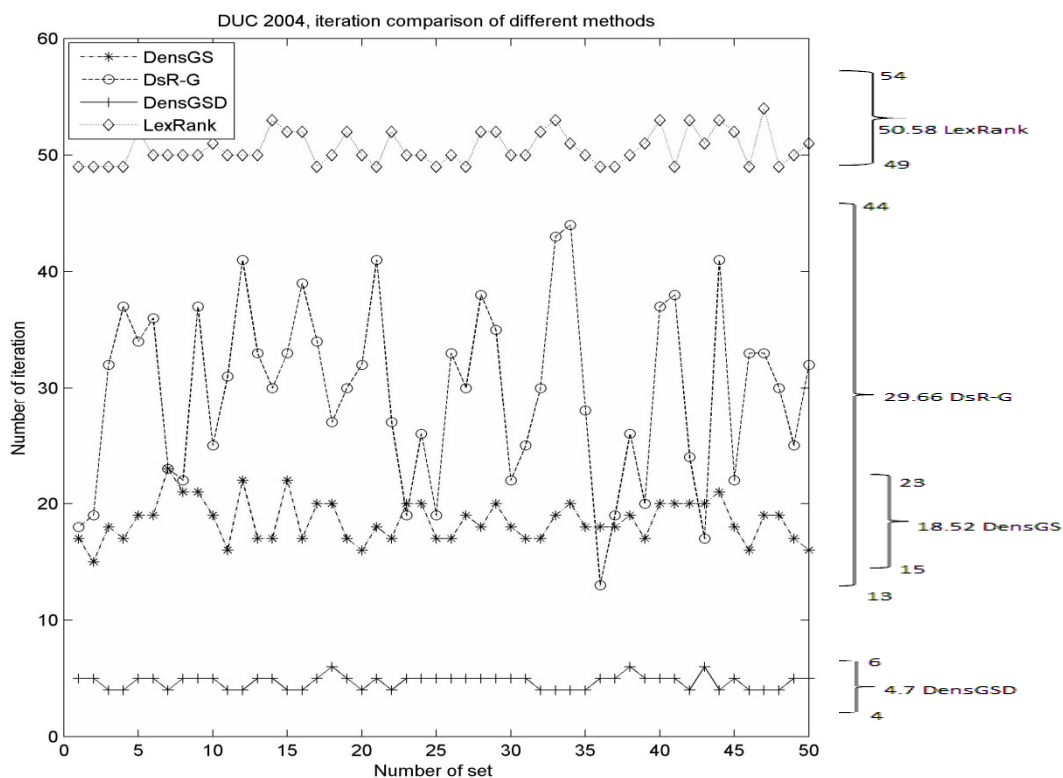


Fig. 4. Iteration comparison of different methods on DUC 2004.

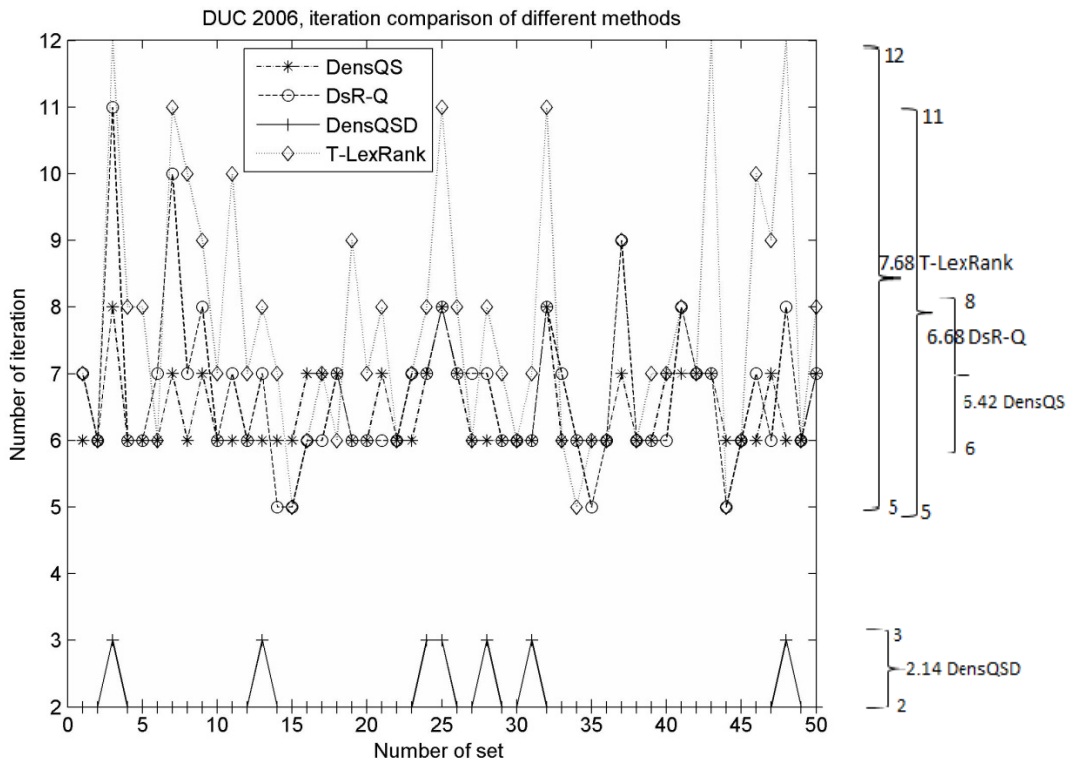


Fig.5. Iteration comparison of different methods on DUC 2006.

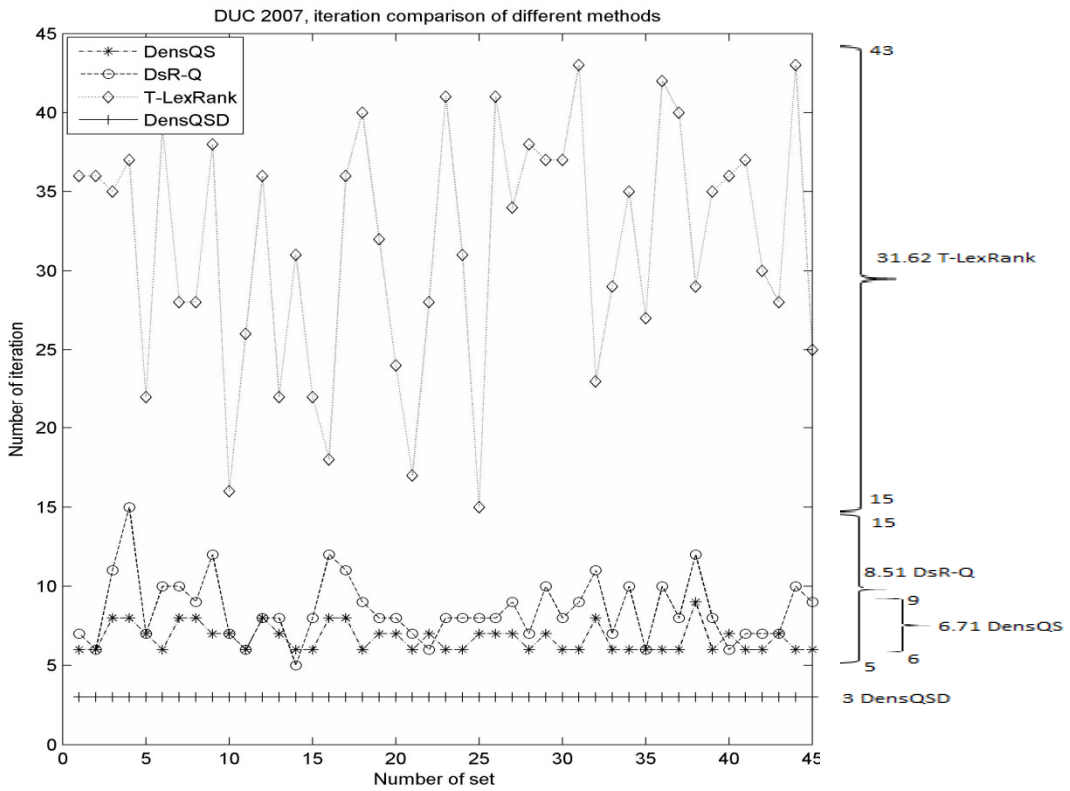


Fig. 6. Iteration comparison of different methods on DUC 2007.

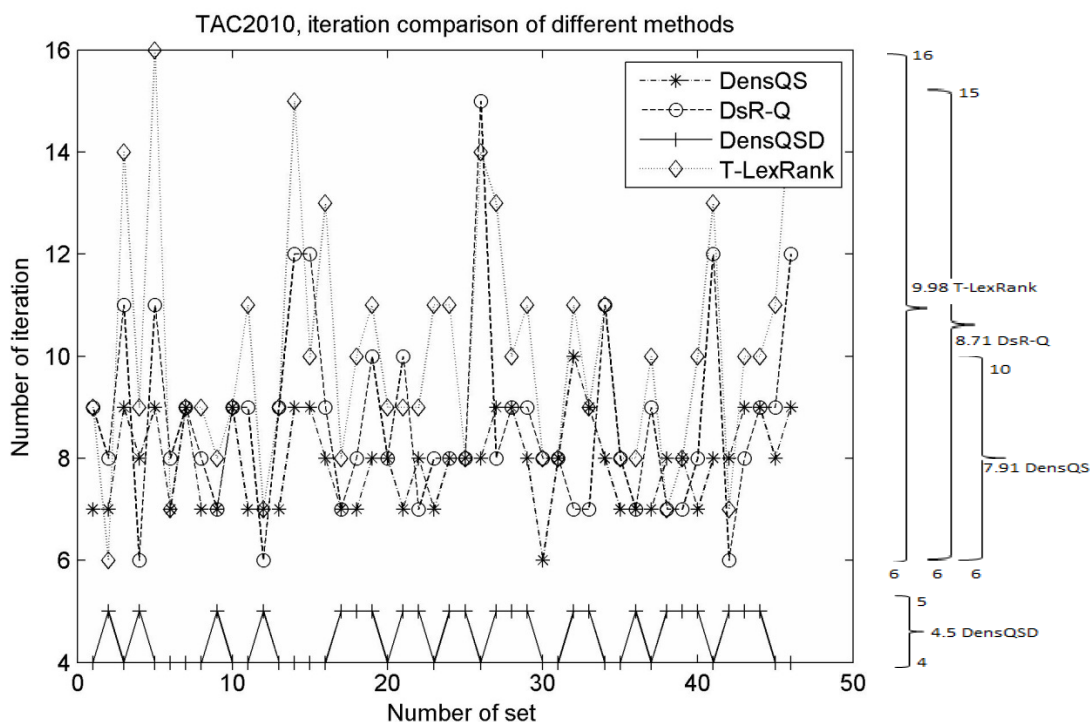


Fig. 7. Iteration comparison of different methods on TAC 2010.

7 Conclusion and Future Work

In this paper we have proposed a new algorithm, graph-based sentence ranking that includes the concept of density both for generic and query-based multi-document summarization. The main contribution of our work is the concept of density that has been added to the graph model to improve the accuracy and reduce the number of iterations. The ROUGE evaluations on four different data sets showed that the density concept can notably improve the results and the method has better performance compared to the algorithm that includes the centroid value. Our algorithms have reached better performance than DsR-G and DsR-Q.

The results confirm that including more information to the algorithms improves their performance. We note that our system ranks 2nd (but not 1st) for some of the data sets (e.g. DUC 2006 and DUC 2007) when compared to other participating systems. This is because the participating systems use syntactical and semantic analysis. Our research just focuses on the effect of the density on the graph-based methods and explores this notion. It shows that this concept (density) is effective and improves the previous version of the graph based algorithms. Therefore, it is conceivable that it can potentially improve other graph-based systems.

Furthermore, the experiments show that our idea reduces the number of iterations significantly. Our system is about 8 times faster than the baseline algorithms (e.g. LexRank and T-LexRank) on average.

In our future work we plan to investigate other methods (e.g. usage of sentence length or supervised methods) to improve further the performance and reduce the number of iterations in graph-based multi-document summarization.

References

- [1] Erkan G., Radev D. (2004). LexRank: Graph-based Lexical Centrality as Saliency in Text Summarization. *Journal of Artificial Intelligence Research (JAIR)* 22 (pp. 457-479).
- [2] Brin S., Page B. (1998). The anatomy of a large-scale hypertextual web search engine. *Comp. Networks ISDN Syst.* 30(1-7) (pp. 107-117).
- [3] Page L., Brin S., Motwani R., Winograd T. (1998). The PageRank citation ranking: bringing order to the web, *Proceedings of the 7th Int. World Wide Web Conference*. Brisbane, Australia (pp. 161-172).
- [4] Wei F., Li W., Lu Q., He Y. (2010). A document-sensitive graph model for multi-document summarization. *Knowledge Information Systems* 22(2) (pp. 245-259).
- [5] Wei F., He Y., Li W., Lu Q. (2008). A Query-Sensitive Graph-Based Sentence Ranking Algorithm for Query-Oriented Multi-document Summarization. *International Symposium on Information Processing ISIP* (pp. 9-13).
- [6] Radev D., Jing H., Sty M., Tam D. (2004). Centroid-based summarization of multiple documents. *Inf. Process. Manage.* 40(6) (pp. 919-938).
- [7] Kleinberg M. (1999). Authoritative sources in hyperlinked environment, *J ACM* 46(5) (pp. 604-632).
- [8] Padmanabhan D., Desikan P., Srivastava J., Riaz K. (2005). WICER: A Weighted Inter-Cluster Edge Ranking for Clustered Graphs. *Web Intelligence* (pp. 522-528).
- [9] Lin C. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of workshop on text summarization, Branches Out*, Post-conference workshop of ACL, Barcelona, Spain.
- [10] Wan X., Yang J., Xiao J. (2006). Using Cross-Document Random Walks for Topic-Focused Multi-Document Summarization. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 1012-1018).
- [11] Zhao L., Wu L., Huang X. (2009). Using query expansion in graph-based approach for query-focused multi-document summarization. *Inf. Process. Manage.* 45(1) (pp. 35-41).
- [12] Haveliwala T. (2002). Topic-sensitive PageRank. *WWW* (pp. 517-526).
- [13] You O., Li W., Zhang R., Li S., Lu Q. (2013). A progressive sentence selection strategy for document summarization. *Inf. Process. Manage.* 49(1) (pp. 213-221).
- [14] Otterbacher J., Erkan G., Radev D. (2005). Using Random Walks for Question-focused Sentence Retrieval. *Human Language Technology Conference and Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP)*.
- [15] Langville AN., Meyer CD., (2004). Deeper inside PageRank. *J Internet Math* 1(3) (pp. 335-380).
- [16] MacCluer CR., (2000). The many proofs and applications of Perron's theorem. *SIAM Rev.* 42(3) (pp. 487-498).
- [17] DUC 2004, <http://www-nlpir.nist.gov/projects/duc/guidelines/2004.html>.
- [18] DUC 2006, <http://www-nlpir.nist.gov/projects/duc/guidelines/2006.html>.
- [19] DUC 2007, <http://duc.nist.gov/duc2007/tasks.html>.
- [20] TAC 2010, <http://www.nist.gov/tac/2010/Summarization/Guided-Summ.2010.guidelines.html>.