Contextual salience in query-based summarization

Wauter Bosma
CLTL, VU University Amsterdam
Boelelaan 1105, 1081HV Amsterdam
w.bosma@let.vu.nl

Abstract

Discourse theories claim that text gets meaning in context. Most summarization systems do not take advantage of this. They assess the relevance of each passage individually rather than modeling the way context affects the relevance of passages. This paper presents a framework for graph-based summarization in order to model relations in text, so that the passages can be viewed in a broader context. The result is a summarization system which is more in line with discourse theory but still fully automatic. I evaluated the content selection performance of an implementation of the framework in different configurations. The system significantly outperforms a competitive baseline (and participant systems) on the DUC 2005 evaluation set.

Keywords

Query-based summarization, content selection, semantic networks, discourse structure, graph theory.

1 Introduction

One of the challenges in automatic summarization is content selection – deciding what should be in a summary, and what shouldn't. Summarization systems typically do this by determining the relevance of each passage independently, and then composing a summary of the top passages. Classical features for scoring sentences include the presence of cue phrases, term frequency, stop word lists, etc. [4, 7].

Systems which assess the relevance of each sentence individually violate insights in discourse organization (e.g., [9]), which claim that meaning is tightly related to discourse organization. The meaning in a text is not merely the sum of the meaning in its passages, but a passage should be interpreted in the context shaped by other passages. For example, given the two passages in Fig. 1, the second passage had little meaning if the context provided by the first would be omitted. Hence, a generic summarization system should include the second sentence in a summary only if the first (or similar) is also included. Recently, summarization systems have broadened their scope from generic single document summarization to multi-document summarization, query-based summarization and update summarization [11]. These summarization tasks have made the need for dealing with issues like redundancy and coherence even more critical. For instance, in case of query-based summarization, the query is part of the summary's context. Update summarization extends the context to prior knowledge, represented by a number of documents which are assumed to be read by the user.

A number of ad-hoc solutions to redundancy and coherence emerged in response to the increasingly complex summarization tasks. For instance, [3] introduced the concept of *marginal relevance*: i.e., that the salience of a content unit is reduced by its redundancy with respect to the summary thus far. [1] divided the source into topics by identifying *lexical chains*. They composed summaries of one sentence from each of the strongest topics, as to maximize coverage. The summarization system of [2] prefers to include sentences in the summary which have a coherence relation to another summary sentence. Each of these answers to the problem of coherence represent a small change to an existing summarization system, rather than a new methodology based on the notion of coherence. Some summarization systems (e.g., [10, 13]) do assign a prominent and explicit role to coherence relations, but they require high level knowledge which can only be annotated manually. A fully automatic graph-based summarization system was built by [5], but their aim was to select sentences which represent a particular (sub)topic in the text, rather than to model coherence or contextual salience.

This paper presents a graph-based framework for content selection in automatic summarization which is based on *contextual salience* – all evidence of salience of a particular passage is based on the salience of related passages (its context). In the evaluation setting, the features used to calculate salience include a graph to express relations between sentences of the same document based on cosine similarity, and a graph to express redundancy, also based on cosine similarity. Section 2 describes the evaluated task and the data set used for evaluation. Section 3 describes the summarization framework. Section 4 describes the experiments to evaluate the framework, and section 5 describes the results.

1A A commercial airliner crashed in northwestern Iran on Wednesday.
1B All 168 people on board were killed.

Fig. 1: *Related passages.*

International Conference RANLP 2009 - Borovets, Bulgaria, pages 39–44

2 Evaluation procedure

The DUC 2006 data set is used in this paper for training, and the DUC 2005 data set is used for testing.[1] This is possible because the data sets for DUC 2005 and DUC 2006 are similar. The task posed by the evaluation set is to automatically generate a summary of a maximum of 250 words, given a *topic*. A topic consists of a title, a query, and a set of source documents. The summary should answer the query, using the source documents. An example of a topic is given in Fig. 2. The DUC 2006 document set consists of 50 topics with 25 source documents each. The DUC 2005 document set consists of 50 topics with 25–50 source documents each (approx. 32 on average).

The summarization task is given to professional human summarizers as well as automatic summarization systems. The human summaries are used as *reference summaries* for evaluating *candidate summaries* (i.e., generated summaries). Each DUC 2005 topic has six corresponding reference summaries; each DUC 2006 topic has four. I use Rouge-2 (i.e. bigram recall with respect to reference summaries) and Rouge-SU4 (skip bigram recall) as performance metrics for evaluation [6], because these metrics are also used (with the same configuration) at DUC 2005 and DUC 2006. Although Rouge metrics provide only a partial evaluation of a summarization system, they are very suitable for these experiments since they require no manual intervention. Other evaluation methods (including extrinsic methods) may be applied at a later stage.

To measure if one summarization algorithm performs better (or worse) than another with a particular metric, I count the number of topics for which it outperformed the other, and vice versa. Then, an approximate randomization test is run to measure statistical significance.

3 A framework for summarization

The aim of this paper is to investigate the content selection sub task of summarization. Nonetheless, the evaluation methods used are designed to measure the quality of abstracts, and require a full summarization system. I briefly describe the summarization system, and then focus on the content selection components. The summarization system consists of the following components.

Segmentation. The source documents as well as the query are segmented into sentences. In addition to the textual content, the document name, the paragraph number and sentence number are associated with each sentence. The document name can be used to detect whether sentences are from the same document, or whether they are query sentences. Paragraph boundaries are derived from annotations provided with the source documents. The segmenter also attempts to remove meta data from the text, such as the date and location of publication. These meta data are not part of the running text and may introduce noise in the summary.

[1] These data are available from `http://duc.nist.gov`

Title:	former President Carter's international activities
Query:	Describe former President Carter's international efforts including activities of the Carter Center.

Fig. 2: *A DUC 2006 topic (D0650E).*

Feature extraction. The source text and the query are processed and converted to a feature graph to prepare for content selection. Multiple modules may be used in parallel so that multiple graphs are generated. This may include coherence analysis, measuring redundancy, etc. The generated graphs are integrated into a combined graph, as described later.

Salience estimation. A salience value is derived for each sentence from the (possibly combined) feature graph.

Presentation. A summary is created using the most salient content units, up to the word limit of 250 words. If adding the next-salient sentence would cause the word limit to be exceeded, no more sentences are added. Where possible, the linear ordering of the sentences in the source text is retained. If the summary contains sentences from multiple source documents, sentences from the document containing the largest number of sentences are presented first. Although the ordering of the sentences may be important for readability, it has little effect on Rouge scores.

The components of *segmentation* and *presentation* remain constant. The experiments described in the next section are used to compare different methods for *feature extraction* and *salience estimation*.

4 Experiments in query-based summarization

This section describes a number of experiments, starting with a rudimentary summarization system, and adding features to build increasingly sophisticated systems. The modular summarization framework allows for the flexibility to add feature graphs or replace the salience estimation algorithm.

The first summarization system, called *query-relevance*, just measures the similarity of candidate sentences with query sentences. The only feature graph – the *query-relevance graph* – relates candidate sentences to query sentences by cosine similarity. The most similar candidate sentences are included in the summary.

Next, a feature graph is added which relates candidate sentences to other sentences of the same document, by means of cosine similarity. This is the *cohesion graph*. Two salience estimation algorithms are used: an adapted version of the *normalized centrality* algorithm, first published in [5], and the *probabilistic relevance* algorithm.

Finally, another feature graph is added – the *redundancy graph* – which relates candidate sentences to sentences of another document, by means of cosine

Table 1: *Performance on DUC 2006 data: Rouge scores, and the system rank among 36 systems (bracketed) if it had participated in DUC 2006.*

System	Rouge-2		Rouge-SU4	
Query-relevance	0.0818	(11)	0.138	(11)
Normalized c.	0.0820	(11)	0.136	(11)
Probabilistic r.	0.0888	(3)	0.143	(7)
Redundancy-aware n.c.	0.0929	(2)	0.150	(2)
Redundancy-aware p.r.	0.0930	(2)	0.150	(2)

Table 2: *Percentage of DUC 2006 topics (Rouge-2/Rouge-SU4) for which one system (rows) beat another (columns). Note that percentages do not add up to 100 if both systems receive the same score for at least one topic. The compared systems are (a) query-relevance (Δ_q); (b) normalized centrality ($\Delta_{q,c}$); (c) probabilistic relevance ($\Delta_{q,c}$); (d) normalized centrality ($\Delta_{q,c,r}$); (e) probabilistic relevance ($\Delta_{q,c,r}$; P_r).*

%	(a)	(b)	(c)	(d)	(e)
(a)	–	50/52	$34^a/28^a$	$30^a/28^a$	$26^a/26^a$
(b)	46/48	–	$34^a/36^b$	$38^b/34^a$	$30^a/24^a$
(c)	$64^a/70^a$	$66^a/62^b$	–	56/58	44/50
(d)	$66^a/66^a$	$60^b/62^a$	42/42	–	$30^a/30^a$
(e)	$70^a/72^a$	$68^a/72^a$	48/46	$64^a/68^a$	–

[a] Significant at $p < 0.01$.
[b] Significant at $p < 0.05$.
[c] Significant at $p < 0.1$.

similarity. This graph can be used in combination with the previously used graphs as well as both salience estimation algorithms.

The remainder of this section describes the summarization systems in greater detail, and gives preliminary comparative performance statistics on DUC 2006 data. Table 1 gives an overview of the Rouge scores of each system. A pair-wise comparison of the systems is shown in Table 2.

4.1 Query-relevance

A simple form of query-based summarization is to determine sentence salience by measuring its cosine similarity with the query. The sentences most similar to the query are presented as a summary. This constitutes a competitive baseline system for query-based summarization. The graph used for salience estimation is the graph where each candidate sentence is related to each query sentence, and the strength of this relation is the cosine similarity of the two sentences. The sentences closest to a query sentence are then included in the summary. The cosine similarity graph is generated in three steps:

1. words of all sentences are stemmed using Porter's stemmer [12];
2. the inverse document frequency (IDF) is calculated for each word;
3. the cosine similarity of each candidate sentence and each query sentence is calculated using the $tf \cdot idf$ weighting scheme.

Stemming is a way to normalize syntactic variation. The inverse document frequency is used to weight words higher than other words if they occur in fewer sentences. Rare words typically characterize the sentence they appear in to a greater extent than frequent words.

Using this method for calculating IDF values for query terms as well appeared not appropriate because there is a mismatch between the language use in the query and in the source documents. For instance, queries frequently used phrases such as 'Discuss ...' or 'Describe ...'. These words have a low frequency in the source documents, and are thus assigned a high IDF value, but they are hardly descriptive if they appear in the query. Therefore, the IDF values for query terms are calculated from the set of sentences from all DUC 2006 queries instead of the source document sentences specific for the topic.

The query-relevance graph (δ_q) is defined by a function determining the strength of the relation between two sentences:

$$\begin{aligned} \delta_q(i,j) &= cosim(i,j) && \text{, if } i \in Q;\ j \in S \\ \delta_q(i,j) &= 0 && \text{, otherwise} \end{aligned} \quad (1)$$

where $\delta_q(i,j)$ is the strength of the relation between sentences i and j; Q is the set of query sentences; S is the set of candidate sentences; $cosim(i,j)$ is the cosine similarity of sentences i and j. The strength of a relation is a value in the range of 0 (no relation) to 1 (a strong relation).

The query-relevance $R_{query\_relevance}(j)$ of a sentence j is then calculated as follows.

$$R_{query\_relevance}(j) = \min_{q \in Q} \delta_q(q,j) \quad (2)$$

where $R_{query\_relevance}(j)$ is the salience of sentence j; Q is the set of query sentences.

A summary is then generated from the most salient sentences. The results are shown in Table 1 and Table 2.

4.2 Contextual relevance

The *cohesion graph* (δ_c) is added as a feature graph for calculating contextual relevance. This graph is constructed indentically to the way the query-relevance graph is constructed, except that it relates candidate sentences of the same document, rather than query sentences and candidate sentences.

The graphs δ_q and δ_c are integrated into a single multi-graph $\Delta_{q,c}$. A multi-graph is a graph that can have two edges between the same two vertices, expressing simultaneous relations. As a result, not a single relation but a set of relations hold between two sentences, and each relation may have a different strength between 0 and 1. The integrated graph is expressed as follows.

$$\Delta_{q,c}(i,j) = \{w_q \delta_q(i,j), w_c \delta_c(i,j)\} \quad (3)$$

where $\Delta_{q,i}(i,j)$ is a set of values, each representing the strength of an edge from i to j in the multi-graph $\Delta_{q,c}$. The values of $w_q, w_c \in [0..1]$ are weighting factors. The smaller w_q and the greater w_c, the greater

the relative importance of indirect evidence of relevance, and the more sentences are selected which are not directly query-relevant.

The salience estimation algorithms calculate the salience of each sentence, given a graph of relations between sentences. A relation from sentence X to sentence Y increases the relevance of Y if X is relevant. This immediately poses a problem if X is a candidate sentence, because initially, its relevance is unknown, and the relevance of Y depends on the relevance of X. Literature provides two solutions [8, 5], both of which iteratively recalculate the salience of a sentence from a similarity graph and the salience of neighboring sentences. Following this process, relevance is calculated as follows.

1. Initiate the salience of all candidate sentences (source document sentences) at 0. The salience of query sentences is initiated at 1.
2. Recalculate the salience of each candidate sentence, using the feature graphs and the salience of neighboring (i.e. related) sentences. Salient sentences increase the salience of their neighbors.
3. Repeat step 2 unless the change in salience in the last iteration falls below a certain (pre-defined) threshold.

I used two salience estimation algorithms, *normalized centrality* and *probabilistic relevance*. They differ in how they recalculate relevance (step 2).

The first, based on [5], recalculates the salience by dividing the salience of each sentence among its neighboring sentences. Because no salience is created or lost (the total 'amount of salience' of all sentences remains approximately constant), I call this *normalized centrality*.

The *probabilistic relevance* algorithm regards the feature graph as a probabilistic semantic network. The salience of a sentence represents the probability that the sentence is relevant, and a relation from sentence X to Y is the probability that Y is relevant, given X is relevant.

Normalized centrality

At each iteration, the normalized centrality is calculated as follows:

$$\begin{aligned} \mu_j(t) &= 1 && \text{, if } j \in Q \\ \mu_j(0) &= 0 && \text{, if } j \in S \\ \mu_j(t+1) &= \frac{d}{\|D\|} + (1-d)\sum_{i \in D} x(i,j) && \text{, if } j \in S \\ x(i,j) &= \sum_{r \in \Delta_{q,c} ij} r \cdot \mu_i(t) \cdot degree(i)^{-1} \end{aligned} \tag{4}$$

where $D = Q \cup S$; and $\mu_j(t)$ is the normalized centrality of sentence j at iteration $t \geq 0$; and $\Delta_{q,c} ij$ is the set of edges between i and j in the relevance graph. The constant d is a small value which is required in generic summarization in order to guarantee a salience ranking under all circumstances by giving each sentence a small prior non-zero salience.[2] The degree of a sentence i in the graph ($degree(i)$) is measured as the number of outgoing edges:

$$degree(i) = \sum_{k \in D} \sum_{(r \in \Delta_{q,c}(i,k))} r \tag{5}$$

The result is a salience value μ between 0 and 1 associated with each passage. The content units with the highest salience values are selected for inclusion in the summary. In this configuration, normalization cancels out the effect of graph weighting: changing the graph weights w_q and w_c (eq. 3) does not affect the summaries in any way because the relevance distribution is normalized and the sets of sentences with outgoing edges in δ_q and δ_c are disjunct.

As shown in Table 2, the average quality of normalized centrality summaries does not significantly differ (at $p < 0.05$) from the quality of query-relevance summarization.

Probabilistic relevance

In the probabilistic approach, contrary to the normalized approach, the relevance of Y given X is unaffected by any other sentence whose relevance may depend on X. Viewing edges as relevance probabilities also has implications on how evidence of relevance is combined. Rather than accumulating weighted relevance of neighbors, the relevance of a sentence is calculated as the product of inverse conditional probabilities. This is based on the idea that, if we have several pieces of evidence that a sentence is salient, it suffices if one of them is true. The probabilistic relevance algorithm calculates salience as follows.

$$\begin{aligned} \nu_j(t) &= 1 && \text{, if } j \in Q \\ \nu_j(0) &= 0 && \text{, if } j \in S \\ \nu_j(t+1) &= 1 - \prod_{(i \in Q \cup S)} z(i,j) && \text{, if } j \in S \\ z(i,j) &= \prod_{r \in \Delta_{q,c}(i,j)} (1 - r \cdot \nu_i(t) \cdot y) \end{aligned} \tag{6}$$

where $\nu_j(t)$ is the probabilistic relevance value of sentence j at iteration t. The value of y is the *decay* value, a global constant in the range $\langle 0..1 \rangle$. The constant y has a function similar to the constant d in normalized centrality: it is necessary to ensure that the salience value keeps increasing at each iteration.

The graph weights w_q and w_c are determined by measuring Rouge-2 performance for different weight values. First, w_q is incremented in steps of 0.1 from 0 to 1 with $w_c = 1$, and then w_c is incremented in steps of 0.1 from 0 to 1 with $w_q = 1$. The optimal weight settings are $w_q = 1$; $w_c = 0.1$ (see Table 1 for Rouge scores). As shown in Table 2, the system significantly outperforms the query-relevance system ($p < 0.01$ for Rouge-2 and Rouge-SU4) and the normalized centrality system ($p < 0.05$ for Rouge-2 and Rouge-SU4).

[2] Throughout this section, the value of 0.15 is used, as suggested in [5], but the actual value of d has no effect on the final salience ranking as long as it is non-zero.

4.3 Redundancy-aware summarization

One of the assumptions usually made implicitly in the design of single-document summarization systems, is that the source document does not contain redundancy. Consequently, there is no risk of including a sentence in the summary which does not contain any information not already present. This changes when a summary is generated from multiple source documents, where non-redundancy of sentences from different documents cannot be taken for granted. The content selection procedures outlined previously concentrate entirely on relevancy, not redundancy. However, in multi-document summarization, presented content should be relevant to the query and novel with respect to what is already mentioned in the summary. In other words, salience comprises both relevance and novelty.

To accommodate representing novelty, the model is extended with a redundancy feature graph P which is used in addition to the previously mentioned relevancy feature graph Δ. Similarly to relevance, redundancy relations have a strength in the range [0..1]. The strength of a redundancy relation between two sentences expresses the likelihood that a sentence is redundant, given the fact that another sentence is redundant. The redundancy of sentence j, given sentence i, is defined by $\delta_r(i,j)$. The form of the redundancy graph is identical to that of the relevance graph. The strengths of relations in the redundancy feature graph δ_r are defined as follows:

$$\begin{aligned} \delta_r(i,j) &= cosim(i,j) && \text{, if } i,j \in S;\ doc(i) \neq doc(j) \\ \delta_r(i,j) &= 0 && \text{, otherwise} \end{aligned} \tag{7}$$

The redundancy-aware summarization system uses a set of redundancy feature graphs P for determining salience of sentences, in addition to the relevancy feature graphs Δ:

$$\begin{aligned} \Delta_{q,c,r}(i,j) &= \{w_q \cdot \delta_q(i,j), w_c \cdot \delta_c(i,j), w_{r\Delta} \cdot \delta_r(i,j)\} \\ P_r(i,j) &= \{w_{rP} \cdot \delta_r(i,j)\} \end{aligned} \tag{8}$$

where $\delta_q(i,j)$, $\delta_c(i,j)$ and $\delta_r(i,j)$ are the query-relevance graph, the cohesion graph, and the redundancy graph respectively. The set of relations between sentences i and j are represented by $\Delta_{q,c,r}(i,j)$ (relevancy) and $P_r(i,j)$ (redundancy). Since redundancy implies 'relatedness', I regard a redundancy graph a special case of a relevance graph. Therefore, δ_r is not only included in P_r but also in $\Delta_{q,c,r}$.

The calculation of redundancy-adjusted salience was inspired by [3]. First, the relevance of each sentence is calculated using $\Delta_{q,c,r}$. Then, the novelty is calculated – novelty is the reciprocal of redundancy. If two sentences are redundant, this affects only the novelty of the less-relevant of the two. The stronger the redundancy relation, the greater the reduction of novelty. Novelty is calculated as follows:

$$\begin{aligned} N(j) &= \prod_{i \in F_j} \prod_{r \in P_r(i,j)} (1 - r \cdot R(i)) \\ F_j &= \{k : S \mid R(k) > R(j)\} \end{aligned} \tag{9}$$

where $N(j)$ is a value in the range [0..1], representing the novelty of sentence j; $P_r(i,j)$ is a set of redundancy relations, expressing the redundancy of j given i; F_j is the set of content units more relevant than j. The function $R(i)$ denotes the relevance of sentence i, as previously calculated.

Now, the redundancy-adjusted salience can be calculated as the product of relevancy and novelty:

$$\sigma_j = R(j) \cdot N(j) \tag{10}$$

where σ_j is the redundancy-adjusted salience of sentence j. The calculation of σ_j ensures that:

- if one content unit is selected, all content units redundant to that unit are less likely to be selected: if two content units are redundant with respect to each other, the salience of the less-relevant content unit is reduced;
- redundancy of a content unit does not prevent relevancy to propagate: a redundant content unit may still be relevant.

The graph weights are determined by starting from the optimal values for w_q and w_c in section 4.2. The remaining weights are determined by means of a similar procedure as in section 4.2: first, $w_{r\Delta}$ is incremented in steps of 0.1 from 0 to 1 with $w_{rP} = 0$, and then w_{rP} is incremented in steps of 0.1 from 0 to 1 without changing the other weights.

For the normalized centrality algorithm, the resulting optimal weight settings are $w_q = 1$; $w_c = 1$ and $w_{rP} = 0$; $w_{r\Delta} = 1$. Increasing the value of $w_{rP} = 0$ has no effect on the quality of the summaries. Table 1 shows the system's performance with these settings on DUC 2006 data. As shown in Table 2, the redundancy-aware normalized centrality system significantly outperforms the normalized centrality system ($p < 0.05$ for Rouge-2 and Rouge-SU4).

For the probabilistic relevance algorithm, the resulting optimal weight settings are $w_q = 1$; $w_c = 0.1$; $w_{r\Delta} = 0.2$; $w_{rP} = 1$. This configuration shows a significant performance gain compared to all previously mentioned systems ($p < 0.01$ for Rouge-2 and Rouge-SU4) except the (non-redundancy aware) probabilistic relevance system. Compared to the latter, the performance was increased but no significant differences were found.

5 Validating the results

The previous section outlined a comparison of different configurations of the summarization framework. However, the way the graph weight configurations are determined implies that the weights are tailored to the DUC 2006 data set. As a result, there is a risk that the weights are overfitted to this particular set. In order to validate the results, I ran the experiments on the DUC 2005 data set with the graph weight configurations determined in section 4.

Fig. 3 shows the average Rouge-2 and Rouge-SU4 scores achieved with the DUC 2005 corpus. Table 3 shows an overview of the pair-wise significance tests. The redundancy-aware probabilistic relevance system significantly outperformed all other systems when Rouge-2 is used ($p < 0.1$), and all except the

Table 3: *Percentage of DUC 2005 topics (Rouge-2/Rouge-SU4) for which one system (rows) beat another (columns). Note that percentages do not add up to 100 if both systems receive the same score for at least one topic. The compared systems are (a) query-relevance (Δ_q); (b) normalized centrality ($\Delta_{q,c}$); (c) probabilistic relevance ($\Delta_{q,c}$); (d) normalized centrality ($\Delta_{q,c,r}$); (e) probabilistic relevance ($\Delta_{q,c,r}$; P_r).*

%	(a)	(b)	(c)	(d)	(e)
(a)	–	46/44	42/42	50/50	$40^{c}/40^{c}$
(b)	52/54	–	$50/34^{a}$	50/54	$38^{b}/34^{a}$
(c)	54/58	$50/66^{a}$	–	$58^{b}/64^{a}$	$36^{c}/42$
(d)	44/44	46/44	$38^{b}/36^{a}$	–	$30^{a}/30^{a}$
(e)	$58^{c}/60^{c}$	$60^{b}/66^{a}$	$54^{c}/54$	$60^{a}/70^{a}$	–

[a] Significant at $p < 0.01$.
[b] Significant at $p < 0.05$.
[c] Significant at $p < 0.1$.

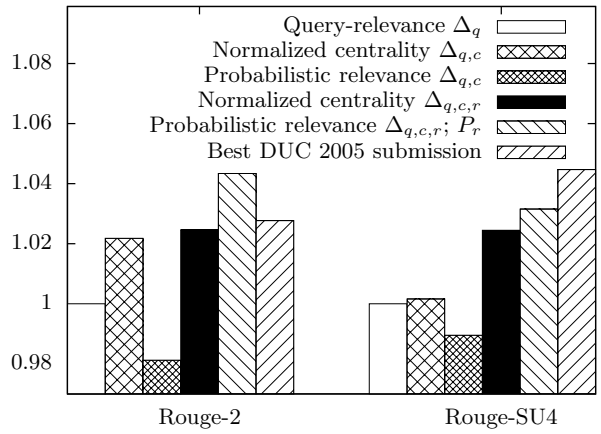

Fig. 3: *Indexed performance on DUC 2005 data:* 1 *indicates the performance of the query-relevance system.*

redundancy-aware normalized centrality system according to Rouge-SU4. This system would have ranked first (Rouge-2) or second (Rouge-SU4) if it had participated in DUC 2005.

Note that it is not guaranteed that the combination of graph weights that leads to the best performance has been found. Apart from the risk of overfitting, the number of possible graph weight combinations is infinite and a greater number of graphs makes it more difficult to find the best combination of weights. A future extension would use machine learning methods such as genetic algorithms to be better suited to find the optimal solution. As mentioned before, Rouge measured only one aspect of a summarization system. That said, the results may teach us the following:

1. The graph-based approach to summarization represents a promising direction, given the good results in spite of the superficial linguistic analysis performed by the evaluated systems. Even better results are to be expected when more sophisticated features are used.
2. The probabilistic interpretation of semantic networks (i.e., *probabilistic relevance*) seems to be more suitable for content selection than the social network interpretation (i.e., *normalized centrality*).

6 Conclusion

The aim of this paper is to bring automatic summarization practice in line with insights from discourse theory. To this end, it provides a framework for automatic summarization which is founded on graph theory. The content selection algorithm is entirely based on relations between text passages. The evaluated system is just one implementation of this framework; it can be extended to exploit more textual features, and discourse oriented features in particular. The framework represents a step toward context aware summarization. Previous work on query-based summarization has mainly focused on extracting the set of sentences which best match the query, ignoring their broader context.

The features used for relating sentences are computationally cheap and easy to port to other languages, but knowledge-intensive methods may detect relations between sentences more accurately. Despite this, the graph-based approach showed good results compared to DUC participant systems (the redundancy-aware probabilistic relevance system would have ranked first for Rouge-2 and second for Rouge-SU4 if it had participated in DUC 2005), which indicates that we are on the right track. Further performance gains may be achieved by using more different sources of information for detecting relations, including knowledge-intensive methods such as rhetorical relation detection or anaphora resolution.

References

[1] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17, Aug. 1997.

[2] S. Blair-Goldensohn and K. McKeown. Integrating rhetorical-semantic relation models for query-focused summarization. In *Proceedings of DUC*, 2006.

[3] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, New York, NY, USA, 1998.

[4] H. Edmundson. New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285, Apr. 1969.

[5] G. Erkan and D. Radev. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*, 2004.

[6] C.-Y. Lin. Rouge: a package for automatic evaluation of summaries. In *Proceedings of the ACL workshop: Text Summarization Branches Out*, Barcelona, Spain, 2004.

[7] H. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[8] I. Mani and E. Bloedorn. Multi-document summarization by graph search and matching. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI'97)*, pages 622–628, 1997.

[9] W. Mann and S. Thompson. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8:243–281, 1988.

[10] D. Marcu. Discourse trees are good indicators of importance in text. In I. Mani and M. Maybury, ed., *Advances in Automatic Text Summarization*, pages 123–136. MIT Press, 1999.

[11] P. Over, H. Dang, and D. Harman. DUC in context. *Information processing and management*, 43(6):1506–1520, 2007.

[12] M. Porter. Snowball: A language for stemming algorithms, 2001. `http://snowball.tartarus.org/texts/introduction.html`.

[13] F. Wolf and E. Gibson. Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 31(2):249–288, 2005.