IADIS International Journal on Computer Science and Information Systems
Vol. 2, No. 1, pp. 18-32
ISSN: 1646-3692

SUMGRAPH: TEXT SUMMARIZATION USING CENTRALITY IN THE PATHFINDER NETWORK

Kaustubh Patil *LIACC-NIAAD, University of Porto, R. Ceuta, 118-6o, Porto, 4050-190, Portugal*

Pavel Brazdil *LIACC-NIAAD/FEP, University of Porto, R. Ceuta, 118-6o, Porto, 4050-190, Portugal*

ABSTRACT

We present a graph theoretic technique for automatic text summarization aimed at producing extractive summaries of a single document. In our system, called as SumGraph, text is represented as a graph with sentences as nodes while weights on the links represent intra-sentence dissimilarity. Novelty of our approach lies in the use of Pathfinder Network Scaling (PFnet) technique representing conceptual organization of the text which in turn is used to compute importance of a sentence in the text. Importance of a sentence is defined using its centrality in the PFnet. Use of Latent Semantic Analysis (LSA) is also investigated. PFnet and LSA have been shown to model human aspects of semantic memory and linguistic acquisition respectively. The system is empirically evaluated on DUC2001 and DUC2002 datasets using ROUGE measure. Results show that SumGraph performs better than other systems, including a commercial summarizer. Use of LSA did not show any improvement in ROUGE score. We also show that SumGraph is statistically different than other methods using a non-parametric statistical test.

KEYWORDS

Text summarization, graph theory, pathfinder network scaling, node centrality

1. INTRODUCTION

Text summarization deals with problem of generating a shorter version of a text while preserving most of the useful information. Traditionally summarization is done by humans, but due to so called *information overload problem* we must seek automatic means of summarization. Automatic text summarization (TS) is a well studied area and recently has gained widespread interest due to overwhelming amount of textual information available in electronic format. TS is the problem of condensing the source text into a shorter version while preserving the information content. TS can be applied to a single document or a cluster of related documents (multi-document). In this work we deal with only single document summarization. Furthermore, TS techniques can be broadly grouped into abstractive summarization and extractive summarization. *Abstractive summarization* relies on Natural Language Processing (NLP) techniques to parse, interpret and generate text. NLP machinery is computationally expensive and far from perfect at the moment. On the other hand, *extractive summarization* is the process of verbatim extraction of textual units (sentences, paragraphs etc.) from the source text. Extractive summarization is easier and faster than abstractive summarization. If the textual unit to be extracted is a sentence (as in this work) then summarization can be viewed as a problem of selection of a subset of sentences from the source text. This is a hard task as the summarization system must somehow understand the text in order to identify the important sentences to be extracted as the summary.

Along another dimension TS methods can be categorized into unsupervised techniques and supervised techniques [Chuang and Yang, 2000; Kupiec et al, 1995]. A drawback of supervised techniques is that they need annotated corpus, which is expensive. Also supervised techniques are not portable; as a summarizer trained for a particular purpose (e.g. language, technical documents) can not be used for other purpose without retraining. On the other hand unsupervised techniques do not rely on annotated corpus and roughly include the following approaches, surface level indicators and corpus statistics [Luhn, 1958; Edmundson, 1969], graph theory based techniques [Salton et al., 1997; Mihalcea, 2004; Erkan and Radev, 2004] and approaches employing natural language processing [Aone et al., 1997].

In general an extractive summarization system follows the framework shown in figure 1. Most of the systems vary in step 2. Some systems use different sentence selection methods in step 3 to improve information coverage, particularly in the multi-document summarization due to potential information redundancy. A comprehensive survey of the field can be found in [Mani and Maybury, 1999].

1. Sentence boundary discrimination
2. Calculation of sentence importance (ranking)
3. Selection of ranked sentences

Figure 1. General framework for extractive summarization

We present a graph theoretic technique to produce summary of a single document, called as SumGraph. Use of graphs in summarization task is not a new idea. In the current work the proposed method is similar to LexRank [Erkan and Radev, 2004]. A novelty of our work is that we use a conceptually sound link reduction technique called as Pathfinder Network Scaling (PFnet) [Schvaneveldt, 1990; Schvaneveldt et al, 1988] and results indicate that our system performs better. The idea is to find salient sentences in the text using a centrality measure. In other words, the central nodes in a PFnet identify the important sentences in the corresponding text. We also investigate use of Latent Semantic Analysis (LSA) [Deerwester et al., 1990] a technique known to capture aspect of human linguistic acquisition [Landauer and Dumais, 1997].

This paper is organized as follows. In subsection 1.1 we shortly discuss evaluation methods. In section 2 we present our approach SumGraph, followed by a brief description of

PFnet. Section 3 discusses datasets used and parameter settings. In section 4 we present results and conclude the work in section 5.

1.1 Summary evaluation

A summary should have several properties; information coverage, coherence, non-redundancy. Assessing usefulness of a summary is a hard task and unfortunately there is no generally accepted automated method available which captures all the properties.

Summary evaluation measures are often grouped along two axes: *extrinsic measures* and *intrinsic measures* [Spark-Jones and Galliers, 1995]. Extrinsic evaluation often involves use of summaries for a specific task, e.g. information retrieval or question answering [Mani, 2001]. While intrinsic evaluation can involve human assessors who grade summaries and/or comparison between automatic summaries and model summaries. Model summaries are often human-written abstracts.

Recently Lin and Hovy [Lin and Hovy, 2003] proposed an intrinsic summary evaluation method called, Recall-Oriented Understudy for Gisting Evaluation (ROUGE). ROUGE uses n-gram statistics to compute usefulness of an automatically generated summary of fixed length by comparing it with gold-standard summaries. In this study we use ROUGE-1 score, i.e. 1-gram matching between automatic summary and model summaries, which is shown to correlate with human judgment [Lin and Hovy, 2003]. High ROUGE score indicates higher relevance of the automatically generated summary and is upper bounded by 1. ROUGE has been used for official evaluation in DUC 2003, DUC 2004 and DUC 2005.

2. SUMGRAPH: PROPOSED METHOD

In this section we discuss in detail how the sentence salience scores are calculated and combined. We follow the general framework in figure 1. In order to get salience of a sentence (step 2 in the framework) we use linear combination of two scores extracted from text. The first score is the centrality of a sentence in the PFnet and the second score is the position weight of a sentence in the text. We first discuss the intuitions behind the use of graph theoretic approach followed by actual calculation part.

2.1 Use of graph theory

In a natural language text the sentences are related to each other. This relatedness can be in the form of lexical overlap. In lexical relatedness two sentences sharing same words are related to each other. We exploit this idea in order to calculate importance of a sentence in the text. Importance of an entity, like a sentence, is not an individual property, but it depends upon other entities. Graphs provide an elite way for representing relationships between entities. Moreover graphs can also be used to calculate *relative importance* of entities by analyzing the graph structure.

In order to use these notions the text should be represented as a graph. A graph is made up nodes and links. The links can be weighted with the weights indicating the strength of relationship between the nodes. Moreover, the links could be directed reflecting the direction of the interaction. In this work we use undirected weighted links. Sentences are used as the nodes of a graph while the links represent lexical similarity between pairs of sentences. Calculation of lexical similarity is discussed in section 2.3. Thus text can be represented as a fully connected graph that is each node is connected with every other node.

Once we obtain a fully connected graph link reduction techniques can be used to prune the graph to contain only critical links. Such a reduced graph uncovers the hidden or latent structure underlying raw data, a fully connected graph. The link reduction algorithms are generally called *scaling algorithms*. The aim of a scaling algorithm is to prune a dense network in order to reveal the latent structure underlying the data which is not visible in the raw data. There are two scaling approaches: threshold-based and topology-based. In t*hreshold-based approach* elimination of a link is solely decided depending upon whether its weight exceeds some threshold. A major disadvantage of a threshold-based scaling algorithm is that the threshold value must be defined. The threshold value can be different for different datasets. A high threshold can lead to a poorly pruned graph while a low threshold can lead to a highly pruned graph. In either case the scaled graph does not provide useful information about node importance. On the other hand a *topology-based approach* eliminates a link considering topological properties of the network. Therefore a topology-based approach preserves intrinsic network properties reliably. The Pathfinder Network Scaling method possesses many advantages over other scaling techniques, such as multidimensional scaling [Chen and Morris, 2003]. Following sub-section describes the Pathfinder Network Scaling algorithm.

2.1.1 Pathfinder Network Scaling

Pathfinder Network Scaling (PFnet) is a topology-based link-reduction technique originally proposed by cognitive scientists [Schvaneveldt et al,, 1988; Schvaneveldt et al., 1989; Schvaneveldt, 1990] for scaling of proximity data (pair-wise relatedness between concepts as perceived by a human subject) to generate a network model of human semantic memory. PFnets are connected networks and are generated by preserving only critical links and removing the other links. Non-critical links are identified by comparing the direct cost (distance) between a pair of nodes with the costs of the alternative paths between them. The direct link is considered as non-critical and removed from the network if the direct cost is greater than the indirect cost; this is known as the *triangular inequality criterion.* The total cost of a path P is calculated using the Minskowski distance;

$$W(P) = \sum_{i=1}^{q} (W_i^r)^{1/r} \qquad (1)$$

where q is the number of links and W_i is the weight on the link i. As it can be seen in equation 1 there are two parameters r and q:

- **r Parameter:** This parameter defines the metric space using the Minskowski distance measure and increasing it amplifies relative contribution of the larger weights to the path. This parameter can assume any value from 1 to ∞. When $r = 2$ we get the well-known Euclidean distance. At $r = \infty$ the Minskowski distance is defined as the maximum weight of its component links.
- **q Parameter:** This parameter is the upper limit on the number of links to be considered to calculate alternate paths between a pair of nodes. q can take any integer value between 2 and n-1, where n is the number of nodes in the network.

The complexity of the scaled network decreases as either r or q increases. The link between nodes i and j is eliminated if:

$$w_{ij} \leq W(P) \qquad (2)$$

where w_{ij} is the weight on the direct link between the corresponding nodes. Thus edge membership is determined depending upon (possibly) all the connections across the entire network. The PFnet(r=∞, q=n-1) is the maximally reduced (least dense) network revealing only the most salient links. Moreover, PFnet(r=∞, q=n-1) is the union of all minimum spanning trees of PFnet(r, q) thus providing a network with least number of links with all minimum cost paths [Chen and Morris, 2003]. In essence the procedure scales a fully connected network in order to get a link-reduced network using the notion of triangle inequality to uncover the latent structure.

This PFnet uncovers the conceptual organization of the sentences in a text. In order to obtain salience of a sentence we analyze the structure of the PFnet using node centrality measures. As each node represents a sentence the important nodes essentially point to the important sentences. In the following sub section we give an overview of the node centrality measures.

2.1.2 Node centrality

We borrow the concept of centrality from social network analysis. The pioneering work was done by Bavelas [Bavelas, 1950] followed by various others. Since then various centrality measures have been proposed. In sociometrics centrality was originated in the pursuit of characterizing importance of an individual or an organization, which is abstractly referred to as an actor. It is well accepted that importance is consequence of relations between the actors and the patterns of connection therefore. Thus the node centrality measures seek for *relative importance* of a node in a graph. Various node centrality measures have been proposed depending upon the definition of centrality. In this work we compare four different centrality scores explained below;

- ***Degree centrality*** identifies the nodes that reach most of the nodes directly. This measure captures local importance of a node.
- ***Closeness centrality*** of a node identifies how easily other nodes can be reached from it. High degree centrality nodes are positioned in the network to quickly diffuse information. This is a global measure.
- ***PageRank*** [Page et al, 1998] is used by Google for ranking web pages. PageRank uses voting to decide importance of a node in the network while importance of the voters is also considered. It is an iterative algorithm.
- ***Eigenvector centrality*** takes into account the importance of neighbors of a node in order to compute its importance. Thus a node which is connected many other important nodes gets high importance. This is also a global measure.

Thus each sentence i in the text gets a score C_i using one of the centrality measures explained above.

2.2 Position heuristic

As this paper deals with newspaper articles we take advantage of the structural aspect of those documents. Newspaper articles are written in such a way that the leading sentences are generally important. Each sentence in the text is assigned a weight as following;

$$P_i = \frac{1}{\sqrt{i}} \tag{3}$$

where i is the position of the sentence in the text. Thus the first sentence gets the highest score equal to 1 while the last sentence gets the lowest score.

Both scores are then scaled between 0-1 and linearly combined to get the final score R_i of a sentence as follows;

$$R_i = W_c \times C_i + W_p \times P_i \tag{4}$$

where W_c and W_p are the weights for centrality score and position score respectively. The sentences are then ranked according to the final scores. Finally, the sentences are successively added to the summary till required size of the summary is met. In the final summary sentences are ordered as they appear in the original text. In the next section we explain the procedure to obtain centrality score using PFnet.

2.3 Computation of sentence centrality

We use the vector space model [Salton et al., 1975] representation of the sentences in a text. Each sentence is converted into a vector of dimensionality equal to the number of distinct words in the text. Porter stemming [Porter, 1980] is used to stem the words reducing dimensionality of the vector space. Each cell in the resulting vector space model contains frequency of occurrence of the particular term in the sentence. We use the well known inverse document frequency (idf) [Sparck-Jones, 1972] global weighting scheme, which decreases importance of the terms present in many sentences. Once the vector space model has been constructed cosine similarity is used to compute pair-wise similarities between sentences. Those similarity values are then converted into dissimilarities by subtracting them from 1. Resulting dissimilarity matrix can be viewed as a fully connected network with sentences as nodes and dissimilarities as link costs. This network is then scaled using PFnet to obtain conceptual organization of the sentences. If two sentences are connected in a PFnet then they are lexically similar to each other. Thus, the central nodes in the PFnet represent important sentences in the text. This centrality can be measured using several centrality measures as discussed in section 2.1.2.

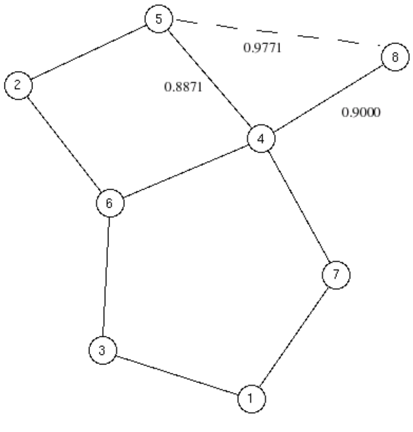
2.4 An example

We take a document, ap880217-0175, from DUC2001 dataset. This document contains 8 sentences; Figure 2 shows the corresponding PFnet and sentence scores. To give an example of link reduction, let's consider direct link between nodes 5 and 8 which is greater than the cost of the links from node 5 to node 4 and from node 4 to node 8. Thus the direct link between node 5 and node 8 violates the triangular inequality criterion and can be removed from the network. As it can be clearly seen, nodes 4 and 6 are highly connected in the network and thus achieve high closeness centrality scores. The 100 word summary for this document includes sentences 1,2, 4 and 6. ROUGE-1 score for this summary is 0.52389.

3. SETTINGS AND EXPERIMENTAL EVALUATION

3.1 Datasets and evaluation

We use newswire data from first two Document Understanding Conferences (DUC), DUC2001 and DUC2002. DUC2001 contains 284 documents from test and training sets and DUC2002 contains 516 documents in total. The task is to produce 100 word summary of each document. Each document is accompanied by 3 human-written summaries in DUC2001 dataset and 2 human-written summaries in DUC2002[1] dataset, which are used as model summaries for evaluation. We use the ROUGE-1[2] as the measure of goodness of an automatically generated summary. For calculating ROUGE-1 score only first 100 words of automatically generated summaries were considered, words were stemmed using Porter stemmer and stop words were not removed.



a) PFnet(r=∞, q=2)

[1] We discard documents in clusters D076 & D098 in DUC2002 as they contain single human-written summary each

[2] ROUGE version 1.1.5 was used

Sentence	*1*	*2*	3	*4*	5	*6*	7	8
Closeness centrality	*0.4667*	*0.4667*	0.5000	*0.7000*	0.5000	*0.6364*	0.5385	0.4375
Position score	*1.0000*	*0.7071*	0.5774	*0.5000*	0.4472	*0.4082*	0.3780	0.3536
Total score	*1.1111*	*0.6580*	0.5843	*1.2265*	0.3830	*0.8422*	0.4224	0.0000
Rank	*2*	*4*	5	*1*	7	*3*	6	8

b) Sentence salience scores (closeness and position scores are not normalized), top three are shown in *italics*

Figure 2. a) PFnet(r=∞, q=2) and b) corresponding sentence salience scores for document ap880217-0175 from DUC2001 dataset

3.2 Other summarization methods

We compare SumGraph with two baselines LEAD and RANDOM, a state-of-art summarizer MEAD [Radev et al., 2001] and a graph based system LexRank [Erkan and Radev, 2004].

The first baseline LEAD takes first n sentences of the source text till required summary size is met. It is well known that LEAD performs quite well on news articles [Brandow et al., 1995] and is a hard to beat baseline. RANDOM system selects n random sentences from the text to fill the required summary size.

MEAD summarizer works in three stages; first stage *feature extractor* extracts features from a document. The second stage *combiner* combines the extracted features into a single scalar value. The third and final stage *reranker* selects the sentences for inclusion in the summary using the score obtained in the combiner stage. The feature extractor by default provides three features; *Centroid, Position* and *Length*. The centroid of a cluster is defined as a pseudo-document which contains the words with a tf*idf value higher than a predefined threshold. The sentences that contain more words from the centroid of the cluster get higher centroid scores. Position is the normalized value of the position of the sentence in the document as defined in equation 3. Length is rather a cut-off value than a feature. The sentences with less number of words than a predefined value are not allowed to enter in the summary. MEAD is a well known open source summarizer and detailed discussion can be found in [Radev et al., 2001].

LexRank [Erkan and Radev, 2004] is a graph-based algorithm that uses threshold-based link reduction and random-walks on graphs to compute salience. LexRank was placed first in more than one tasks in DUC 2004 evaluation. The graph representation is similar to the one used in this work with sentences as the nodes and the intra-sentence cosine similarity as the weights on the links between sentences. The links are then pruned using a user-defined threshold on cosine similarity value. This results in a link reduced sub-graph which in turn is used to calculate salience of the sentences using eigenvector centrality.

We use MEAD and LexRank in three different settings; MEAD – uses centroid feature, MEAD+LexRank - uses centroid in conjunction with LexRank feature and LexRank – uses LexRank feature. All of those three settings use position feature and all features are combined using equal weight.

3.3 Parameters and settings

Sentence length cut-off of 10 was used for all systems, meaning the sentences that contain less than 10 words are discarded. This value was decided empirically on a subset of DUC2001 dataset. For LexRank link reduction threshold of 0.15 is used. We use identity reranker for MEAD and LexRank based systems.

A major drawback of PFnet is its polynomial computational complexity. Construction of a PFnet is an iterative process which starts at q=2 and ends at q=n-1. At q=n-1 we obtain maximally reduced PFnet (for details see [Schvaneveldt, 1990]). We define a simple heuristic as, if there is no reduction in the number of links for 5 consecutive iterations then the iterations are stopped and the resulting network is considered as the maximally reduced network. Moreover, we compared maximally reduced networks for DUC2001 data (defined with the heuristic) with PFnet(r=∞, q=2) networks and observed that almost all of them are identical. So we use q=2 and r=∞ for all experiments.

We tried optimizing the weights in equation 4 (for SumGraph) using a real coded genetic algorithm (GA) [Goldberg, 1989] with elitism. We performed experiments on DUC2001 dataset with the weights constrained between 0 and 1 within two decimal places. Although GA gave a better combination of weights, the improvement was not considerable and we decided to use same weight, equal to 1, for both scores.

4. RESULTS AND DISCUSSIONS

Before presenting the results of the system's performance we show inter-human agreement. To calculate inter-human agreement one of the human written summaries was treated as a computer generated summary and the rest of the summaries were used to evaluate it's score. Each of the human-written summary was given a chance to be a computer generated summary and the results show mean value, e.g. for DUC2001 dataset there are 3 human-written summaries for each document so the results are the mean value of three evaluations (see table 1).

Table 1. Inter-human agreement

Dataset	Mean ROUGE-1 Score
DUC2001	0.45778
DUC2002	0. 50649

The values in table 1 can be seen as upper-bounds, although the bound is not very tight. The loss in the ROUGE score can be attributed to two factors, subjectivity and change of vocabulary. The former corresponds to the fact that different people (subjects) consider same parts of the text to be of unequal importance. The later factor (i.e. change of vocabulary) corresponds to the use of different words to describe same content.

4.1 Individual features and centrality score

In order to get an idea about various scores we performed experiments using only one feature at a time. Table 2 shows performance of individual features on both datasets. Here *Position* corresponds to the scores of the sentences obtained using equation 3. *Closeness Centrality* corresponds to the closeness centrality values of sentences in the corresponding PFnet. *Centroid* is the feature extracted by MEAD and *LexRank feature* is the scores obtained solely by the LexRank algorithm. As the table clearly points out when used individually Position is the best feature while centrality places 3rd and 2nd on DUC2001 and DUC2002 datasets respectively.

Table 2. Performance of individual features

Dataset	**Position**	**Closeness Centrality**	**Centroid**	**LexRank feature**
DUC2001	0.44391	0.42073	0.41423	0.44548
DUC2002	0.47133	0.45141	0.44093	0.43223

We have observed that for SumGraph *closeness centrality* performs better than other centrality measures mentioned in section 2.1.2. Table 3 shows the comparison of the centrality measures on DUC2001 dataset. The results indicate that closeness centrality score performs better than the other scores. Closeness centrality is the inverse of the mean geodesic distance between a node and all other nodes reachable from it. Thus a node with high closeness centrality is positioned in the network to quickly diffuse information. Although [Erkan and Radev, 2004] point out that eigenvector centrality is a good measure, we note that the technique used to obtain the reduced graph is different.

Table 3. Comparison of centrality measures on DUC2001 dataset

Centrality measure	**ROUGE-1 score**
Closeness	*0.4544*
Eigenvector	0.4539
Degree	0.4526
PageRank	0.4506

4.2 System performance

In this section we empirically compare the systems using ROUGE-1 score. ROUGE-1 score was computed by comparing automatically generated summaries against the model summaries. For RANDOM system we conducted five independent experiments and the mean value is reported here. We did not perform experiments on DUC2002 dataset using Copernic[3].

[3] Experiments on DUC2002 dataset using Copernic were not performed due to impracticality.

As can be seen from table 4 SumGraph consistently performs better than other methods. Besides, for DUC2001 dataset Copernic is the second best system, while LexRank performs better than MEAD and MEAD+LexRank. For DUC2002 data surprisingly LexRank does not perform well and drops to the second last position. On DUC2002 dataset MEAD jumps to the second position after SumGraph. RANDOM is the worst system on both datasets, as expected.

Table 4. Comparison of ROUGE-1 scores, we do not show confidence level for the RANDOM system

a) DUC2001 dataset

Method	**ROUGE-1**	**95% confidence interval**
SumGraph	*0.45435*	*0.44403 - 0.46388*
Copernic	0.45109	0.44094 - 0.46192
LexRank	0.45064	0.44012 - 0.46076
MEAD	0.44773	0.43685 - 0.45858
MEAD+ LexRank	0.44677	0.43640 - 0.45693
LEAD	0.44391	0.43348 - 0.45436
RANDOM	0.39016	0.38147 - 0.40114

b) DUC2002 dataset

Method	**ROUGE1**	**95% confidence interval**
SumGraph	*0.48415*	*0.47682-0.49201*
MEAD	0.47293	0.46534-0.48011
MEAD+LexRank	0.47166	0.46407-0.47932
LEAD	0.47133	0.46351-0.47928
LexRank	0.46988	0.46266-0.47713
RANDOM	0.41865	0.41104-0.42670

4.3 Detailed comparison

To get better understanding of the systems (excluding RANDOM) we compared them using win-tie-loss tables. In a win-tie-loss table two systems A and B are compared along three dimensions; 1. *win*, which indicates how many times the system A beats the system B, 2. *tie*, indicating how many times both systems perform equal and 3. *loss*, shows how many times system A looses to system B. We compare the systems with LEAD, i.e. we keep LEAD as system B and as system A we use one of the rests, results are shown in table 5. Table 5 shows win-tie-loss table with LEAD as the reference system. From table 5, it is very clear that SumGraph performs better that all others along all the three dimensions. Neither MEAD not LexRank show consistent performance. Similar results were obtained when compared with other systems indicating SumGraph is the best system in the current context.

Table 5. Win-tie-loss table with LEAD as the reference system

a) DUC2001 dataset

System	**Win**	**Tie**	**Loss**
SumGraph	158	27	99
Copernic	151	8	125
LexRank	150	13	121
MEAD+LexRank	147	14	123
MEAD	124	27	133

b) DUC2002 dataset

System	**Win**	**Tie**	**Loss**
SumGraph	284	48	184
MEAD+LexRank	264	13	239
MEAD	260	42	214
LexRank	244	24	248

To assess the similarity (or difference) between the methods we used *Wilcoxon paired signed rank test*, which is a non-parametric test for related samples. The null hypothesis is that the two systems compared are the same. Results of this test are shown in table 6. According to the statistics in table 6 at 95% confidence level, SumGraph is statistically different (rejecting the null hypothesis with p-value less than 0.05) than all other systems except for DUC2001 dataset it is similar to LexRank and MEAD+LexRank with p-value of 0.237 and 0.058 respectively. MEAD, LexRank and MEAD+LexRank systems are not statistically different than LEAD summarizer on both datasets. They are also similar to each other. As for the Copernic summarizer it is not similar to any other system.

Table 6. System comparison using Wilcoxon paired signed ranks test, in systems column X – Y reads as system X compared with system Y

Systems	**p-value**	
	DUC2001	**DUC2002**
SumGraph - LEAD	0.000	0.000
SumGraph - MEAD	0.016	0.000
SumGraph - MEAD+LexRank	0.058	0.000
SumGraph - LexRank	0.237	0.000
Copernic - LEAD	0.099	--
MEAD - MEAD+LexRank	0.915	0.974
LexRank - MEAD	0.294	0.435
Copernic - MEAD	0.207	--
MEAD+LexRank - LexRank	0.101	0.310
Copernic - MEAD+LexRank	0.213	--
Copernic - LexRank	0.926	--
LEAD - MEAD	0.508	0.220
MEAD+LexRank - LEAD	0.248	0.433
LexRank - LEAD	0.075	0.987
SumGraph - Copernic	0.112	--

4.4 Use of Latent Semantic Analysis

LSA [Deerwester et al., 1990] is a corpus based statistical technique to uncover the semantic relationships between words. LSA handles synonymy and polysemy by considering word co-occurrence (word context) statistics. LSA places the words and documents that are closely related semantically near each other, i.e. the documents are placed closer even though they don’t implicitly contain the same words. LSA takes term-document (here term-sentence) matrix as input and a information-spreading technique known as Singular Value Decomposition (SVD) is applied to the matrix. SVD finds the dimensions which close to each other based on co-occurrence and then compresses them onto one composite dimension. We investigated the question, whether using LSA can improve performance of our system? It should be noted that we do not use LSA for indexing but rather embed LSA into our existing method SumGraph as an intermediate stage, before calculating sentence similarity and call the resulting system SumGraph+LSA.

One very important question associated with LSA is what should be the reduced dimension? We performed experiments on DUC2001 dataset by varying percentage of variance retained. Results show that system performance improves with increasing variance retained as shown in figure 3. Retaining 95% variance was the best option attaining ROUGE-1 score of 0.453 with 95% confidence interval [0.4437 – 0.4627]. Results show that LSA gives no added advantage, at least in terms of ROUGE-1 score, however it performs better than other systems. We did not perform experiments on DUC2002 dataset using LSA. The inability of LSA to improve performance can be attributed to the lack of data due to small document sizes.

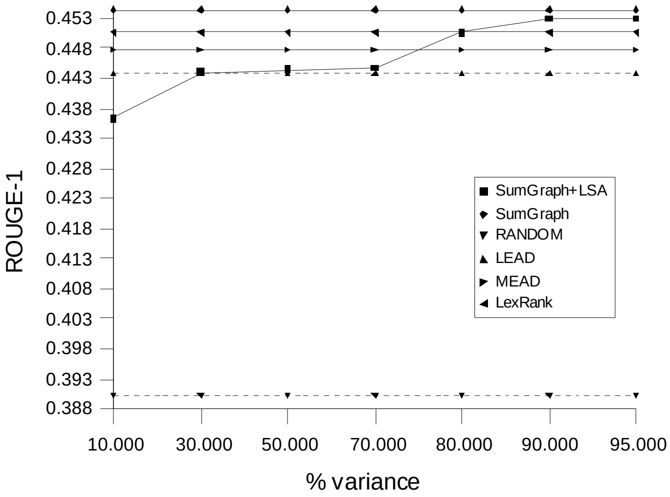


Figure 3. Performance of SumGraph+LSA with changing variance.

5. CONCLUSIONS AND FUTURE WORK

In this work we presented a graph-theory based extractive text summarization algorithm for single documents, called SumGraph. SumGraph is based on Pathfinder Network Scaling (PFnet) which is a conceptually sound systematic link-reduction algorithm. The text is first represented as a fully connected weighted graph. The link weights are calculated using lexical similarity between a pair of sentences. This fully connected graph is then scaled using PFnet algorithm. PFnet reveals conceptual organization of the sentences which is then used to assess salience of sentences using a centrality measure. Amongst the four compared centrality measures; degree, closeness, pagerank and eigenvector, closeness centrality performed better than the others. Nodes with high closeness centrality score are positioned in the network to quickly diffuse information. SumGraph consistently outperforms the other systems when compared using ROUGE-1 score. Performance of either MEAD or LexRank was not consistent. Statistical analysis shows that SumGraph is different than the baseline system LEAD, while the other systems are similar to LEAD. Moreover, we investigated the issue of using latent semantic analysis (LSA). Experiments show that LSA does not show any improvement in the terms of ROUGE-1 score which can be attributed to lack of data.

As future research in this area we intend to evaluate our system for longer summaries. Extension to multi-document summarization is another potential direction. There are some unexplored areas like how the use of co-reference resolution affects the resulting summary. Currently we are working on producing cohesive summaries. It will be interesting to study if LSA plays any role in obtaining cohesive summaries.

ACKNOWLEDGEMENT

The Portuguese Pluri-annual support provided by FCT and funding under the FCT project SUMO is gratefully acknowledged. Authors are thankful to Dragomir Radev (Uni. of Michigan, USA) for MEAD toolkit. KP would like to thank Kiran Patil (DTU, Denmark) and Pedro Campos (LIACC/FEP, Uni. of Porto, Portugal) for various fruitful discussions.

REFERENCES

Aone, C., Okurowski, M.E., Gorlinsky, J., and Larsen, B., 1997. A Scalable Summarization System Using Robust NLP, *Proceeding of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pp. 66-73.

Bavelas, A., 1950. Communication patterns in task-oriented groups. *Journal of the Acoustical Society of America*, 22 (6), pp. 723-730.

Brandow, R., Mitze, K., and Rau, L.F., 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 31(5), pp. 675-685.

Chen, C. and Morris, S., 2003. Visualizing Evolving Networks: Minimum Spanning Trees versus Pathfinder Networks. *INFOVIS*.

Chuang, W. T., and Yang, J., 2000. Extracting sentence segments for text summarization: A machine learning approach. *In Proceedings of the 23rd ACM SIGIR*, pp. 152-159.

Deerwester, S., Dumais, S.T., Furna, G.W., et al., 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science.*

Edmundson, H. P., 1969. New methods in automatic abstracting, *Journal of Association for Computing Machinery,* 16(2), pp. 264-285.

Erkan G., and Radev, D. R., 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *J. Artif. Intell. Res. (JAIR)*, 22, pp. 457-479.

Goldberg, David E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Pub. Co.

Kupiec, J., Pedersen, J. O., and Chen, F., 1995. A trainable document summarizer. *In Research and Development in Information Retrieval,* pp. 68–73.

Landauer, T.K., and Dumais, S.T., 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104, pp. 211-240.

Lin., C.Y., and Hovy, E., 2003. Automatic evaluation of summaries using n-gram co-ocuurrence. *In Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May 27 – June 1.

Luhn, H.P., 1958. Automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2), pp. 159-165.

Mani, I., 2001. Summarization evaluation: An overview. *In proceedings of the NAACL 2001 workshop on Automatic Summarization*,.

Mani, I., and Maybury, M.T., 1999. *Advances in automatic text summarization.* MIT Press.

Mihalcea, R., 2004. Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization. *In Proceedings of the 42nd Annual Meeting of the ACL, companion volume (ACL 2004)*, Barcelona, Spain.

Porter , M., 1980. An algorithm for suffix stripping. *Program*, 14(3), pp. 130-137.

Page, L., Brin, S., Motwani, R., and Winograd, T., The pagerank citation ranking: Bringing order into web, *Technical report, Stanford University*, Stanford, CA, 1998.

Radev, D., Blair-Goldstein, S., and Zang, Z., 2001. Experiments in single and multi-document summarization using MEAD. *In first Document Understanding Conference*, New Orleans, LA.

Salton, G., Wong, A., and Yang, C., 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18, pp. 613-620.

Salton, G., Singhal, A., Mitra, M., and Buckley, C., 1997. Automatic text structuring and summarization. *Information processing & management*, 33(2), pp. 193-207.

Schvaneveldt, R. (Ed.), 1990. *Pathfinder Associative Networks: Studies in Knowledge Organization.* Norwood, NJ, Ablex.

Schvaneveldt, R. W., Dearholt, D. W., & Durso, F. T., 1988. Graph theoretic foundations of Pathfinder networks. *Computers and Mathematics with Applications*, 15, pp. 337-345.

Sparck-Jones, K., 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation,* 28(1), pp. 11-20.

Sparck-Jones, K. and J. R. Galliers. 1995. Evaluating Natural Language Processing Systems: An Analysis and Review. *In Lecture Notes in Artificial Intelligence*, *Springer.*