Multi-Document Summarization via Sentence-Level Semantic Analysis and Symmetric Matrix Factorization

Dingding Wang, Tao Li
School of Computer Science
Florida International University
Miami, FL 33199
dwang003,taoli@cs.fiu.edu

Shenghuo Zhu
NEC Labs. America, Inc
10080 N. Wolfe Rd. SW3-350
Cupertino, CA 95014
zsh@sv.nec-labs.com

Chris Ding
Department of CSE
University of Texas at Arlington
Arlington, TX 76019
chqding@uta.edu

ABSTRACT

Multi-document summarization aims to create a compressed summary while retaining the main characteristics of the original set of documents. Many approaches use statistics and machine learning techniques to extract sentences from documents. In this paper, we propose a new multi-document summarization framework based on sentence-level semantic analysis and symmetric non-negative matrix factorization. We first calculate sentence-sentence similarities using semantic analysis and construct the similarity matrix. Then symmetric matrix factorization, which has been shown to be equivalent to normalized spectral clustering, is used to group sentences into clusters. Finally, the most informative sentences are selected from each group to form the summary. Experimental results on DUC2005 and DUC2006 data sets demonstrate the improvement of our proposed framework over the implemented existing summarization systems. A further study on the factors that benefit the high performance is also conducted.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text clustering*

General Terms

Algorithms, Experimentation, Performance

Keywords

Multi-document summarization, Sentence-level semantic analysis, Symmetric non-negative matrix factorization

1. INTRODUCTION

Multi-document summarization is the process of generating a generic or topic-focused summary by reducing documents in size while retaining the main characteristics of the original documents [21, 27]. Since one of the problems of data overload is caused by the fact that many documents share the same or similar topics, automatic multi-document summarization has attracted much attention in recent years. With the explosive increase of documents on the Internet, there are various summarization applications. For example, the informative snippets generation in web search can assist users in further exploring [31], and in a Qestion/Answer system, a question-based summary is often required to provide information asked in the question [14]. Another example is short summaries for news groups in news services, which can facilitate users to better understand the news articles in the group [28].

The major issues for multi-document summarization are as follows [32]: first of all, the information contained in different documents often overlaps with each other, therefore, it is necessary to find an effective way to merge the documents while recognizing and removing redundancy. In English to avoid repetition, we tend to use different word to describe the same person, the same topic as a story goes on. Thus simple word-matching types of similarity such as cosine can not faithfully capture the content similarity. Also the sparseness of words between similar concepts make the similarity metric uneven. Another issue is identifying important difference between documents and covering the informative content as much as possible [25]. Current document summarization methods usually involve natural language processing and machine learning techniques [29, 2, 34], such as classification, clustering, conditional random fields (CRF) [4], etc. Section 2 will explicitly discuss these existing methods.

In this paper, to address the above two issues, we propose a new framework based on sentence-level semantic analysis (SLSS) and symmetric non-negative matrix factorization (SNMF). Since SLSS can better capture the relationships between sentences in a semantic manner, we use it to construct the sentence similarity matrix. Based on the similarity matrix, we perform the proposed SNMF algorithm to cluster the sentences. The standard non-negative matrix factorization(NMF) deals with a rectangular matrix and is thus not appropriate here. Finally we select the most informative sentences in each cluster considering both internal and external information. We conduct experiments on DUC2005 and DUC2006 data sets, and the results show the effectiveness of our proposed method. The factors that benefit the high performance are further studied.

The rest of the paper is organized as follows. Section 2 discusses the related work of current methods in multi-document

SIGIR'08, July 20–24, 2008, Singapore.

summarization. Our proposed method including SLSS and SNMF algorithm is described in Section 3. Various experiments are set up and the results are shown in Section 4. Section 5 concludes.

2. RELATED WORK

Multiple document summarization has been widely studied recently. In general, there are two types of summarization: extractive summarization and abstractive summarization [16, 15]. Extractive summarization usually ranks the sentences in the documents according to their scores calculated by a set of predefined features, such as term frequency-inverse sentence frequency (TF-ISF) [26, 20], sentence or term position [20, 33], and number of keywords [33]. Abstractive summarization involves information fusion, sentence compression and reformulation [16, 15]. In this paper, we study sentence-based extractive summarization.

Gong et al. [12] propose a method using latent semantic analysis (LSA) to select highly ranked sentences for summarization. [11] proposes a maximal marginal relevance (MMR) method to summarize documents based on the cosine similarity between a query and a sentence and also the sentence and previously selected sentences. MMR method tends to remove redundancy, however the redundancy is controlled by a parameterized model which actually can be automatically learned. Other methods include NMF-based topic specific summarization [24], CRF-based summarization [29], and hidden Markov model (HMM) based method [5]. Some DUC2005 and DUC2006 participants achieve good performance such as Language Computer Corporation (LCC) [1], that proposes a system combining the question-answering and summarization system and using k-Nearest Neighbor clustering based on cosine similarity for the sentence selection. In addition, some graph-ranking based methods are also proposed [22, 9]. Most of these methods ignore the dependency syntax in the sentence level and just focus on the keyword co-occurrence. Thus the hidden relationships between sentences need to be further discovered. The method proposed in [13] groups sentences based on the semantic role analysis, however the work does not make full use of clustering algorithms.

In our work, we propose a new framework based on sentence-level semantic analysis (SLSS) and symmetric non-negative matrix factorization (SNMF). SLSS can better capture the relationships between sentences in a semantic manner and SSNF can factorize the similarity matrix to obtain meaningful groups of sentences. Experimental results demonstrate the effectiveness of our proposed framework.

3. THE PROPOSED METHOD

3.1 Overview

Figure 1 demonstrates the framework of our proposed approach. Given a set of documents which need to be summarized, first of all, we clean these documents by removing formatting characters. In the similarity matrix construction phase, we decompose the set of documents into sentences, and then parse each sentence into frame(s) using a semantic role parser. Pairwise sentence semantic similarity is calculated based on both the semantic role analysis [23] and word relation discovery using WordNet [10]. Section 3.2 will describe this phase in detail. Once we have the pairwise sentence similarity matrix, we perform the symmetric matrix factorization to group these sentences into clusters in the second phase. Full explanations of the proposed SNMF algorithm will be presented in section 3.3. Finally, in each cluster, we identify the most semantically important sentence using a measure combining the internal information (e.g., the computed similarity between sentences) and the external information (e.g., the given topic information). Section 3.4 will discuss the sentence selection phase in detail. These selected sentences finally form the summary.

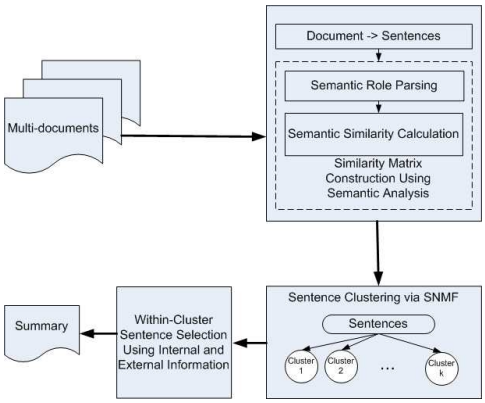

Figure 1: Overview of our proposed method

3.2 Semantic Similarity Matrix Construction

After removing stemming and stopping words, we trunk the documents in the same topic into sentences. Simple word-matching types of similarity such as cosine can not faithfully capture the content similarity. Also the sparseness of words between similar concepts make the similarity metric uneven. Thus, in order to understand the semantic meanings of the sentences, we perform semantic role analysis on them and propose a method to calculate the semantic similarity between any pair of sentences.

3.2.1 Semantic Role Parsing

A semantic role is "a description of the relationship that a constituent plays with respect to the verb in the sentence" [3]. Semantic role analysis plays very important role in semantic understanding. The semantic role labeler we use in this work is based on PropBank semantic annotation [23]. The basic idea is that each verb in a sentence is labeled with its propositional arguments, and the labeling for each particular verb is called a "frame". Therefore, for each sentence, the number of frames generated by the parser equals to the number of verbs in the sentence. There is a set of abstract arguments indicating the semantic role of each term in a frame. For example, Arg0 is typically the actor, and Arg1 is the thing acted upon. The full representation of the abstract arguments [23] and an illustrative example are shown in Table 1.

3.2.2 Pairwise Semantic Similarity Calculation

Given sentence S_i and S_j, now we calculate the similarity between them. Suppose S_i and S_j are parsed into frames respectively. For each pair of frames $f_m \in S_i$ and $f_n \in S_j$, we discover the semantic relations of terms in the same semantic role using WordNet [10]. If two words in the same semantic role are identical or of the semantic relations such

rel: the verb	Arg0: causer of motion
Arg1: thing in motion	Arg2: distance moved
Arg3: start point	Arg4: end point
Arg5: direction	ArgM-LOC: location
ArgM-EXT: extent	ArgM-TMP: time
ArgM-DIS: discourse connectives	ArgM-PNC: purpose
ArgM-ADV: general-purpose	ArgM-MNR: manner
ArbM-NEG: negation marker	ArgM-DIR: direction
ArgM-MOD: modal verb	ArgM-CAU: cause
Example:	
Sentence: A number of marine plants are harvested commercially in Nova Scotia.	
Label: A|Arg1 number|Arg1 of|Arg1 marine|Arg1 plants|Arg1 are|- harvested|rel commercially|ArgM-MNR in|ArgM-LOC Nova|ArgM-LOC Scotia|ArgM-LOC .|-	

Table 1: Representation of arguments and an illustrative example.

as synonym, hypernym, hyponym, meronym and holonym, the words are considered as "related".

Let R_m and R_n be the semantic roles in f_m and f_n, respectively. Assume $R_m \leq R_n$. Let $\{r_1, r_2, ..., r_k\}$ be the set of K common semantic roles between f_m and f_n, $T_m(r_i)$ be the term set of f_m in role r_i, and $T_n(r_i)$ be the term set of f_n in role r_i. Let $|T_m(r_i)| \leq |T_n(r_i)|$, then we compute the similarity between $T_m(r_i)$ and $T_n(r_i)$ as:

$$rsim(T_m(r_i), T_n(r_i)) = \frac{\sum_j tsim(t_{ij}^m, r_i)}{|T_n(r_i)|} \quad (1)$$

where

$$tsim(t_{ij}^m, r_i) = \begin{cases} 1, & t_{ij}^m \in T_m(r_i),\ \exists t_{ik}^n \in T_n(r_i) \\ & s.t.\ t_{ij}^m \text{ and } t_{ik}^n \text{ are related.} \\ 0, & \text{else} \end{cases}$$

Then the similarity between f_m and f_n is

$$fsim(f_m, f_n) = \frac{\sum_{i=1}^k rsim(T_m(r_i), T_n(r_i))}{R_n} \quad (2)$$

Therefore, the semantic similarity between S_i and S_j can be calculated as follows.

$$Sim(S_i, S_j) = \max_{f_m \in S_i, f_n \in S_j} fsim(f_m, f_n) \quad (3)$$

Each similarity score is between 0 and 1. Thus, we compute pairwise sentence similarity for the given document collection and construct the symmetric semantic similarity matrix for further analysis.

3.3 Symmetric Non-negative Matrix Factorization (SNMF)

Most document clustering algorithms deal with a rectangular data matrix (e.g., document-term matrix, sentence-term matrix) and they are not suitable for clustering pairwise similarity matrix. In our work, we propose the SNMF algorithm to conduct the clustering in the second phase. It can be shown that the simple symmetric nonnegative matrix factorization approach is equivalent to normalized spectral clustering.

3.3.1 Problem Formulation and Algorithm Procedure

Given a pairwise similarity matrix W, we want to find H such that

$$\min_{H \geq 0} J = ||W - HH^T||^2. \quad (4)$$

where the matrix norm $||X||^2 = \sum_{ij} X_{ij}^2$ is the Frobenius norm. To derive the updating rule for Eq.(4) with non-negative constraints, $h_{ij} \geq 0$, we introduce the Lagrangian multipliers λ_{ij} and let $L = J + \sum_{ij} \lambda_{ij} H_{ij}$. The first order KKT condition for local minima is

$$\frac{\partial L}{\partial H_{ij}} = \frac{\partial J}{\partial H_{ij}} + \lambda_{ij} = 0, \text{ and } \lambda_{ij} H_{ij} = 0, \forall i, j.$$

Note that $\frac{\partial J}{\partial H} = -4WH + 4HH^TH$. Hence the KKT condition leads to the fixed point relation:

$$(-4WH + 4HH^TH)_{ij} H_{ij} = 0 \quad (5)$$

Using gradient descent method, we have

$$H_{ij} \leftarrow H_{ij} - \epsilon_{ij} \frac{\partial J}{\partial H_{ij}} \quad (6)$$

Setting $\epsilon_{ij} = \frac{H_{ij}}{(8HH^TH)_{ij}}$, we obtain the NMF style multiplicative updating rule for SNMF:

$$H_{ij} \leftarrow \frac{1}{2}[H_{ij}(1 + \frac{(WH)_{ij}}{(HH^TH)_{ij}}] \quad (7)$$

Hence, the algorithm procedure for solving SNMF is: given an initial guess of H, iteratively update H using Eq.(7) until convergence. This gradient descent method will converge to a local minima of the problem.

3.3.2 Properties of SNMF

SNMF has several nice properties that make it a powerful tool for clustering. First, one of the nice properties of the SNMF algorithm is its inherent ability for maintaining the near near-orthogonality of H. Note that

$$||H^TH||^2 = \sum_{st} (H^TH)_{st}^2 = \sum_{s \neq t} (\mathbf{h}_s^T \mathbf{h}_t)^2 + \sum_t (\mathbf{h}_t^T \mathbf{h}_t)^2$$

Minimizing the first term is equivalent to enforcing the orthogonality among $\mathbf{h}_s : \mathbf{h}_s^t \mathbf{h}_t \approx 0$. On the other hand, since $W \approx HH^T$,

$$\sum_{ij} w_{ij} = \sum_{ij} (HH^T)_{ij} = \sum_{s=1}^K |\mathbf{h}_s|^2$$

where $|\mathbf{h}|$ is the L_1 norm of vector $\mathbf{h}$. Hence $||\mathbf{h}_s|| \geq 0$. Therefore, we have

$$\mathbf{h}_s^T \mathbf{h}_t = \begin{cases} 0, & \text{if } s \neq t \\ \|\mathbf{h}_s\|^2, & \text{if } s = t \end{cases}$$

The near-orthogonality of columns of H is important for data clustering. An exact orthogonality implies that each row of H can have only one non-zero element, which leads to the hard clustering of data objects (i.e., each data object belongs to only 1 cluster). On the other hand, a non-orthogonality of H does not have a cluster interpretation. The near-orthogonality conditions of SNMF allow for "soft clustering", i.e., each object can belong fractionally to multiple clusters. This usually leads to clustering performance improvement [7].

Another important property is that the simple SNMF is equivalent to sophisticated normalized cut spectral clustering. Spectral clustering is a principled and effective approach for solving Normalized Cuts [30], a NP-hard optimization problem. Given the adjacent matrix W of a graph, it can be easily seen that the following SNMF

$$\min_{H^TH=I, H\geq 0} ||\widetilde{W} - HH^T||^2. \quad (8)$$

where

$$\widetilde{W} = D^{-1/2}WD^{-1/2},\ D = \text{diag}(d_1, \cdots, d_n),\ d_i = \sum_j w_{ij}.$$

is equivalent to Normalized Cut spectral clustering.

3.3.3 Discussions and Relations

It can also be shown that SNMF is equivalent to Kernel K-means clustering and is a special case of 3-factor Nonnegative matrix factorization. These results validate the clustering ability of SNMF.

Kernel K-means Clustering: For clustering and classification problems, the solution is represented by K nonnegative cluster membership indicator matrix: $H = (\mathbf{h}_1, \cdots, \mathbf{h}_K)$, where

$$\mathbf{h}_k = (0, \cdots, 0, \overbrace{1, \cdots, 1}^{n_k}, 0, \cdots, 0)^T / n_k^{1/2} \quad (9)$$

For example, the nonzero entries of $\mathbf{h}_1$ indicate the data points belonging to the first cluster. The objective function of K-means clustering is

$$J = \sum_{k=1}^{\kappa} \sum_{i \in C_k} \|\mathbf{x}_i - \mathbf{f}_k\|^2 \quad (10)$$

where $\mathbf{f}_k$ is the cluster centroid of the k-th cluster C_k of n_k points, i.e., $\mathbf{f}_k = \sum_{i \in C_k} \mathbf{x}_i / n_k$. More generally, the objective function of Kernel K-means with mapping $\mathbf{x}_i \to \phi(\mathbf{x}_i)$ is

$$J_\phi = \sum_{k=1}^{\kappa} \sum_{i \in C_k} \|\phi(\mathbf{x}_i) - \bar{\phi}_k\|^2 \quad (11)$$

where $\bar{\phi}_k$ is the centroid in the feature space. Using cluster indicators, for K-means and Kernel K-means, the clustering problem can be solved via the optimization problem

$$\max_{H^TH=I,\ H\geq 0} \text{Tr}(H^TWH), \quad (12)$$

where H is the cluster indicator and $W_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ is the kernel. For K-means, $\phi(\mathbf{x}_i) = \mathbf{x}_i$, $W_{ij} = \mathbf{x}_i^T\mathbf{x}_j$.

Note that if we impose the orthogonality constraint on H, then

$$\begin{aligned} J_1 &= \arg\min_{H^TH=I, H\geq 0} ||W - HH^T||^2 \\ &= \arg\min_{H^TH=I, H\geq 0} ||W||^2 - 2\text{Tr}(H^TWH) + ||H^TH||^2 \\ &= \arg\max_{H^TH=I, H\geq 0} \text{Tr}(H^TWH) \end{aligned}$$

In other words, SNMF of $W = HH^T$ is equivalent to Kernel K-means clustering under the orthogonality constraints on H.

Nonnegative Matrix Factorization (NMF): SNMF can also be viewed as a special case of 3-factor nonnegative matrix factorizations. The 3-factor nonnegative matrix factorization is proposed to simultaneously cluster the rows and the columns of the input data matrix X [8]

$$X \approx FSG^T. \quad (13)$$

Note that S provides additional degrees of freedom such that the low-rank matrix representation remains accurate while F gives row clusters and G gives column clusters. This form gives a good framework for simultaneously clustering the rows and columns of X [6, 18]. An important special case is that the input X contains a matrix of pairwise similarities: $X = X^T = W$. In this case, $F = G = H, S = I$. This reduces to the SNMF:

$$\min_{H\geq 0} \|X - HH^T\|^2,\ \ s.t.\ H^TH = I.$$

3.4 Within-Cluster Sentence Selection

After grouping the sentences into clusters by the SNMF algorithm, in each cluster, we rank the sentences based on the sentence score calculation as shown in Eqs.(15, 16, 17). The score of a sentence measures how important a sentence is to be included in the summary.

$$Score(S_i) = \lambda F_1(S_i) + (1-\lambda)F_2(S_i) \quad (15)$$

$$F_1(S_i) = \frac{1}{N-1} \sum_{S_j \in C_k - S_i} Sim(S_i, S_j) \quad (16)$$

$$F_2(S_i) = Sim(S_i, T) \quad (17)$$

where $F_1(S_i)$ measures the average similarity score between sentence S_i and all the other sentences in the cluster C_k, and N is the number of sentences in C_k. $F_2(S_i)$ represents the similarity between sentence S_i and the given topic T. λ is the weight parameter.

4. EXPERIMENTS

4.1 Data Set

We use the DUC2005 and DUC2006 data sets to test our proposed method empirically, both of which are open benchmark data sets from Document Understanding Conference (DUC) for automatic summarization evaluation. Each data set consists of 50 topics, and Table 2 gives a brief description of the two data sets. The task is to create a summary of no more than 250 words for each topic to answer the information expressed in the topic statement.

	DUC2005	DUC2006
Number of topics	50	50
Number of documents relevant to each topic	25 $\sim$ 50	25
Data source	TREC	AQUAINT corpus
Summary length	250 words	250 words

Table 2: Description of the data sets

4.2 Implemented Summarization Systems

In order to compare our methods, first of all, we implement four most widely used document summarization baseline systems:

- **LeadBase**: returns the leading sentences of all the documents for each topic.

- **Random**: selects sentences randomly for each topic.
- **LSA**: conducts latent semantic analysis on terms by sentences matrix as proposed in [12].
- **NMFBase**: performs NMF on terms by sentences matrix and ranks the sentences by their weighted scores [17].

For better evaluating our proposed method, we also implement alternative solutions for each phase of the summarization procedure as listed in Table 3.

Phase	Proposed method	Alternative 1	Alternative 2
Similarity Measurement	Semantic Similarity (SLSS)	Keyword-based similarity	N/A
Clustering Algorithm	SNMF	K-means (KM)	NMF
Within-Cluster Sentence Ranking	$M_p = \lambda F_1(S_i)$ $+(1-\lambda)F_2(S_i)$	$M_1 = F_1(S_i)$	$M_2 = F_2(S_i)$

Table 3: Different methods implemented in each phase. Remark: S_i is the i^{th} sentence in the cluster, and the calculation of $F_1(S_i)$ and $F_2(S_i)$ is the same as described in section 3.4.

In Table 3, the keyword-based similarity between any pair of sentences is calculated as the cosine similarity. The parameter λ in M_p is set to 0.7 empirically, and the influence of λ will be discussed in Section 4.4.4. Note that in our experiments, both similarity matrix generation phase and sentence extraction phase use the same type of similarity measurements. Thus, we have 22 implemented summarization systems: 18 by varying methods in each phase, and 4 baselines. In section 4.4, we will compare our proposed method with all the other systems.

4.3 Evaluation Metric

We use ROUGE [19] toolkit (version 1.5.5) to measure our proposed method, which is widely applied by DUC for performance evaluation. It measures the quality of a summary by counting the unit overlaps between the candidate summary and a set of reference summaries. Several automatic evaluation methods are implemented in ROUGE, such as ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-SU. ROUGE-N is an n-gram recall computed as follows.

$$ROUGE-N = \frac{\sum_{S\in\{ref\}}\sum_{gram_n\in S} Count_{match}(gram_n)}{\sum_{S\in\{ref\}}\sum_{gram_n\in S} Count(gram_n)} \quad (18)$$

where n is the length of the n-gram, and ref stands for the reference summaries. $Count_{match}(gram_n)$ is the maximum number of n-grams co-occuring in a candidatae summary and the reference summaries, and $Count_(gram_n)$ is the number of n-grams in the reference summaries. ROUGE-L uses the longest common subsequence (LCS) statistics, while ROUGE-W is based on weighted LCS and ROUGE-SU is based on skip-bigram plus unigram. Each of these evaluation methods in ROUGE can generate three scores (recall, precision and F-measure). As we have similar conclusions in terms of any of the three scores, for simplicity, in this paper, we only report the average F-measure scores generated by ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-W and ROUGE-SU to compare our proposed method with other implemented systems.

4.4 Experimental Results

4.4.1 Overall Performance Comparison

First of all, we compare the overall performance between our proposed method (using SLSS and SNMF) and all the other implemented systems. Table 4 and Table 5 show the ROUGE evaluation results on DUC2006 and DUC2005 data sets respectively. We clearly observe that our proposed method achieves the highest ROUGE scores and outperforms all the other systems. In section 4.4.2, 4.4.3 and 4.4.4, we evaluate each phase of our proposed method and try to analyze all the factors that our method benefits from.

4.4.2 Evaluation on Methods in Similarity Matrix Construction

Actually, instead of using similarity matrix, many summarization methods directly perform on the terms by sentences matrix, such as the LSA and NMFBase which are implemented as baseline systems in our experiments. In fact, LSA and NMF give continuous solutions to the same K-means clustering problem [7]. Their difference is the constraints: LSA relaxes the non-negativity of H, while NMF relaxes the orthogonality of H. In NMFbase or LSA, we treat sentence as vectors and clustering them using cosine similarity metric (since each document is normalized to 1, $|d1-d2|^2 = 2-2cos(d1,d2)$). From Table 4 and 5, we can see the results of LSA and NMFbase are similar and all of these methods are not satisfactory. This indicates that simple word-matching types of similarity such as cosine can not faithfully capture the content similarity.

Therefore, we further analyze the sentence-level text and generate pairwise sentence similarity. In the experiments, we compare the proposed sentence-level semantic similarity with the traditional keyword-based similarity calculation. In order to better understand the results, we use Figure 2 and 3 to visually illustrate the comparison. Due to space limit, we only show ROUGE-1 results in these figures.

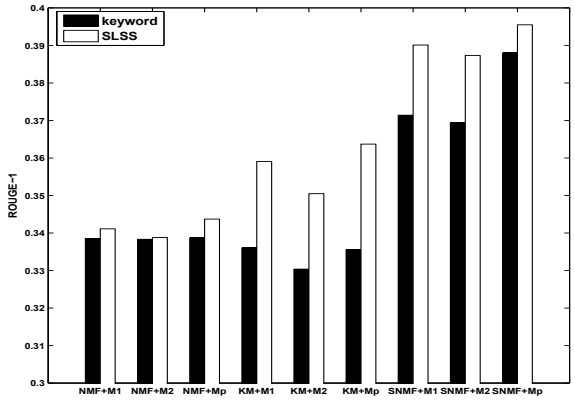

Figure 2: Methods comparison in similarity matrix construction phase using ROUGE-1 on DUC2006 data set

Systems	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W	ROUGE-SU
Average-Human	**0.45767**	**0.11249**	**0.42340**	**0.15919**	**0.17060**
DUC2006 Average	0.37959	0.07543	0.34756	0.13001	0.13206
LeadBase	0.32082	0.05267	0.29726	0.10993	0.10408
Random	0.31749	0.04892	0.29384	0.10779	0.10083
NMFBase	0.32374	0.05498	0.30062	0.11341	0.10606
LSA	0.33078	0.05022	0.30507	0.11220	0.10226
$KM + M_1$ (keyword)	0.33605	0.05481	0.31204	0.12450	0.11125
$KM + M_2$ (keyword)	0.33039	0.04689	0.30394	0.11240	0.10087
$KM + M_p$ (keyword)	0.33558	0.05920	0.32112	0.12614	0.11229
$KM + M_1$ (SLSS)	0.35908	0.06087	0.34074	0.12861	0.12328
$KM + M_2$ (SLSS)	0.35049	0.05931	0.33202	0.12801	0.11763
$KM + M_p$ (SLSS)	0.36371	0.06182	0.34114	0.12966	0.12503
$NMF + M_1$ (keyword)	0.33850	0.05851	0.31274	0.12637	0.11348
$NMF + M_2$ (keyword)	0.33833	0.05087	0.31260	0.11570	0.10662
$NMF + M_p$ (keyword)	0.33869	0.05891	0.31286	0.12719	0.11403
$NMF + M_1$ (SLSS)	0.34112	0.05902	0.32016	0.12951	0.11623
$NMF + M_2$ (SLSS)	0.33882	0.05897	0.31374	0.11650	0.10938
$NMF + M_p$ (SLSS)	0.34372	0.05941	0.32386	0.12973	0.11706
$SNMF + M_1$ (keyword)	0.37141	0.08147	0.35946	0.13214	0.13032
$SNMF + M_2$ (keyword)	0.36934	0.07527	0.34192	0.13011	0.12962
$SNMF + M_p$ (keyword)	0.38801	0.08304	0.36103	0.13361	0.13187
$SNMF + M_1$ (SLSS)	0.39012	0.08352	0.36218	0.13802	0.13713
$SNMF + M_2$ (SLSS)	0.38734	0.08295	0.36052	0.13416	0.13664
$SNMF + M_p$ (SLSS)	**0.39551**	**0.08549**	**0.36803**	**0.13943**	**0.13981**

Table 4: Overall performance comparison on DUC2006 using ROUGE evaluation methods. Remark: "Average-Human" is the average results of summaries constructed by human summarizers and "DUC2006 Average" lists the average results of the 34 participating teams. The system names are the combinations of the methods used in each phase. For example, "$KM + M_2$ (keyword)" represents that keyword-based similarity, K-means clustering and M_2 ranking measurement are used. Candidate methods for each phase are listed in Table 3.

Systems	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W	ROUGE-SU
Average-Human	**0.44170**	**0.10236**	**0.40632**	**0.15227**	**0.16221**
DUC2005 Average	0.34347	0.06024	0.31296	0.11675	0.11488
LeadBase	0.29243	0.04320	0.27089	0.10046	0.09303
Random	0.29012	0.04143	0.26395	0.09802	0.09066
NMFBase	0.31107	0.04932	0.28716	0.10785	0.10094
LSA	0.30461	0.04079	0.26476	0.10883	0.09352
$KM + M_1$ (keyword)	0.31762	0.04938	0.29107	0.10806	0.10329
$KM + M_2$ (keyword)	0.30891	0.04902	0.28975	0.10373	0.09531
$KM + M_p$ (keyword)	0.31917	0.04964	0.29763	0.10856	0.10538
$KM + M_1$ (SLSS)	0.31966	0.05129	0.29882	0.10870	0.10623
$KM + M_2$ (SLSS)	0.31857	0.04971	0.29186	0.10737	0.10594
$KM + M_p$ (SLSS)	0.32149	0.05083	0.29910	0.10886	0.10741
$NMF + M_1$ (keyword)	0.32026	0.05105	0.30012	0.11278	0.10921
$NMF + M_2$ (keyword)	0.32003	0.05086	0.30117	0.11063	0.10205
$NMF + M_p$ (keyword)	0.32218	0.05146	0.30213	0.11605	0.10628
$NMF + M_1$ (SLSS)	0.32943	0.05177	0.30529	0.11743	0.10772
$NMF + M_2$ (SLSS)	0.32231	0.05014	0.30357	0.11918	0.10753
$NMF + M_p$ (SLSS)	0.32949	0.05241	0.30835	0.11807	0.10992
$SNMF + M_1$ (keyword)	0.33118	0.05615	0.31529	0.11903	0.11141
$SNMF + M_2$ (keyword)	0.33025	0.05573	0.31247	0.11839	0.11023
$SNMF + M_p$ (keyword)	0.33402	0.05712	0.31773	0.11918	0.11453
$SNMF + M_1$ (SLSS)	0.34856	0.05909	0.32574	0.11987	0.11641
$SNMF + M_2$ (SLSS)	0.34309	0.05960	0.32381	0.11816	0.11427
$SNMF + M_p$ (SLSS)	**0.35006**	**0.06043**	**0.32956**	**0.12266**	**0.12298**

Table 5: Overall performance comparison on DUC2005 using ROUGE evaluation methods.

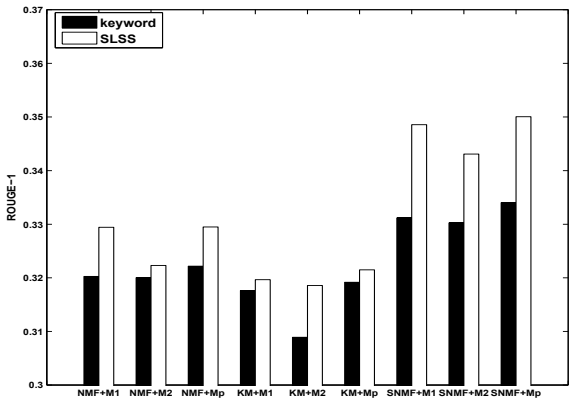

Figure 3: Methods comparison in similarity matrix construction phase using ROUGE-1 on DUC2005 data set

The results clearly show that no matter which methods are used in other phases, SLSS outperforms keyword-based similarity. This is due to the fact that SLSS better captures the semantic relationships between sentences.

4.4.3 *Evaluation on Different Clustering Algorithms*

Now we compare different clustering algorithms in Figure 4 and 5. We observe that our proposed SNMF algorithm achieves the best results. K-means and NMF methods are generally designed to deal with a rectangular data matrix and they are not suitable for clustering the similarity matrix. Our SNMF method, which has been shown to be equivalent normalized spectral clustering, can generate more meaningful clustering results based on the input similarity matrix.

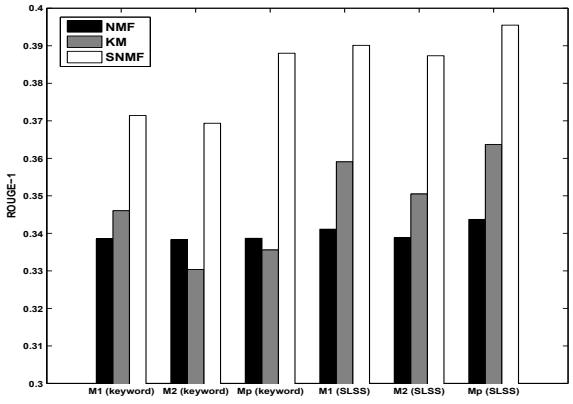

Figure 4: Different clustering algorithms using ROUGE-1 on DUC2006 data set

4.4.4 *Discussion on Parameter λ*

Figure 6 and Figure 7 demonstrate the influence of the weight parameter λ in the within-cluster sentence selection phase. When $\lambda = 1$ (it is actually method M_1), only internal information counts, i.e. the similarity between sentences. And $\lambda = 0$ represents that only the similarity between the sentence and the given topic is considered (method M_2). We

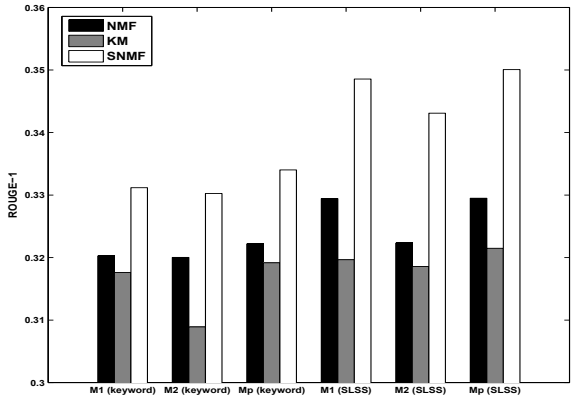

Figure 5: Different clustering algorithms using ROUGE-1 on DUC2005 data set

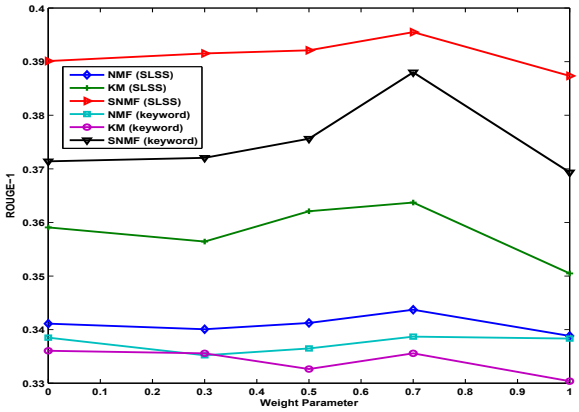

Figure 6: Study of weight parameter λ using ROUGE-1 on DUC2006 data set

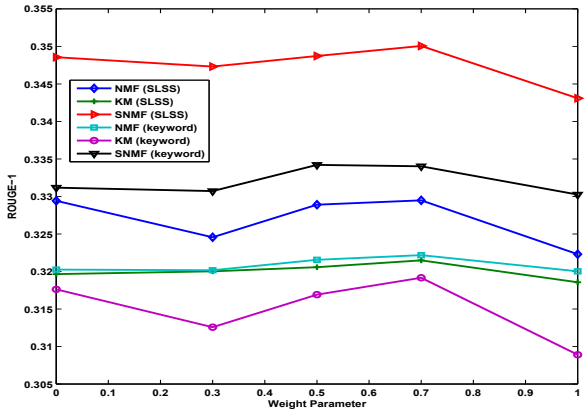

Figure 7: Study of weight parameter λ using ROUGE-1 on DUC2005 data set

gradually adjust the value of λ, and the results show that combining both internal and external information leads to better performance.

5. CONCLUSIONS

In this paper, we propose a new multi-document summarization framework based on sentence-level semantic analysis (SLSS) and symmetric non-negative matrix factorization (SNMF). SLSS is able to capture the semantic relationships between sentences and SNMF can divide the sentences into groups for extraction. We conduct experiments on DUC2005 and DUC2006 data sets using ROUGE evaluation methods, and the results show the effectiveness of our proposed method. The good performance of the proposed framework benefits from the sentence-level semantic understanding, the clustering over symmetric similarity matrix by the proposed SNMF algorithm, and the within-cluster sentence selection using both internal and external information.

Acknowledgements

The project is partially supported by a research grant from NEC Lab, NSF CAREER Award IIS-0546280, and IBM Faculty Research Awards.

6. REFERENCES

[1] http://www-nlpir.nist.gov/projects/duc/pubs/.

[2] M. Amini and P. Gallinari. The use of unlabeled data to improve supervised learning for text summarization. In *Prodeedings of SIGIR 2002*.

[3] D. Arnold, L. Balkan, S. Meijer, R. Humphreys, and L. Sadler. *Machine Translation: an Introductory Guide*. Blackwells-NCC, 1994.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[5] J. Conroy and D. O'Leary. Text summarization via hidden markov models. In *Proceedings of SIGIR 2001*.

[6] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of KDD 2001*.

[7] C. Ding and X. He. K-means clustering and principal component analysis. In *Prodeedings of ICML 2004*.

[8] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of KDD 2006*.

[9] G. Erkan and D. Radev. Lexpagerank: Prestige in multi-document text summarization. In *Proceedings of EMNLP 2004*.

[10] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[11] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of SIGIR 1999*.

[12] Y. Gong and X. Liu. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of SIGIR 2001*.

[13] S. Harabagiu and F. Lacatusu. Topic themes for multi-document summarization. In *Prodeedings of SIGIR 2005*.

[14] T. Hirao, Y. Sasaki, and H. Isozaki. An extrinsic evaluation for question-biased text summarization on qa tasks. In *Prodeedings of NAACL 2001 workshop on Automatic Summarization*.

[15] H. Jing and K. McKeown. Cut and paste based text summarization. In *Prodeedings of NAACL 2000*.

[16] K. Knight and D. Marcu. Summarization beyond sentence extraction: a probablistic approach to sentence compression. *Artificial Intelligence*, pages 91–107, 2002.

[17] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS 2001*.

[18] T. Li. A general model for clustering binary data. In *Proceedings of SIGKDD 2005*, pages 188–197.

[19] C.-Y. Lin and E.Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of NLT-NAACL 2003*.

[20] C.-Y. Lin and E. Hovy. From single to multi-document summarization: A prototype system and its evaluation. In *Proceedings of ACL 2002*.

[21] I. Mani. *Automatic summarization*. John Benjamins Publishing Company, 2001.

[22] R. Mihalcea and P. Tarau. A language independent algorithm for single and multiple document summarization. In *Proceedings of IJCNLP 2005*.

[23] M. Palmer, P. Kingsbury, and D. Gildea. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, pages 71–106, 2005.

[24] S. Park, J.-H. Lee, D.-H. Kim, and C.-M. Ahn. Multi-document summarization based on cluster using non-negtive matrix factorization. In *Proceedings of SOFSEM 2007*.

[25] D. Radev, E. Hovy, and K. Mckeown. Introduction to the special issue on summarization. *Computational Linguistics*, pages 399–408, 2002.

[26] D. Radev, H. Jing, M. Stys, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing and Management*, pages 919–938, 2004.

[27] B. Ricardo and R. Berthier. *Modern information retrieval*. ACM Press, 1999.

[28] G. Sampathsampath and M. Martinovic. *A Multilevel Text Processing Model of Newsgroup Dynamics*. 2002.

[29] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen. Document summarization using conditional random fields. In *Proceedings of IJCAI 2007*.

[30] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[31] A. Turpin, Y. Tsegay, D. Hawking, and H. Williams. Fast generation of result snippets in web search. In *Proceedings of SIGIR 2007*.

[32] X. Wan, J. Yang, and J. Xiao. Manifold-ranking based topic-focused multi-document summarization. In *Proceedings of IJCAI 2007*.

[33] W.-T. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. Multi-document summarization by maximizing informative content-words. In *Proceedings of IJCAI 2007*.

[34] H. Zha. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Prodeedings of SIGIR 2005*.