Multi-Document Summarization Based on Two-Level Sparse Representation Model

He Liu, Hongliang Yu, Zhi-Hong Deng*
Key Laboratory of Machine Perception (Ministry of Education),
School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
lhdgriver@gmail.com, yuhongliang324@gmail.com, zhdeng@cis.pku.edu.cn

Abstract

Multi-document summarization is of great value to many real world applications since it can help people get the main ideas within a short time. In this paper, we tackle the problem of extracting summary sentences from multi-document sets by applying sparse coding techniques and present a novel framework to this challenging problem. Based on the data reconstruction and sentence denoising assumption, we present a two-level sparse representation model to depict the process of multi-document summarization. Three requisite properties is proposed to form an ideal reconstructable summary: **Coverage**, **Sparsity** and **Diversity**. We then formalize the task of multi-document summarization as an optimization problem according to the above properties, and use simulated annealing algorithm to solve it. Extensive experiments on summarization benchmark data sets DUC2006 and DUC2007 show that our proposed model is effective and outperforms the state-of-the-art algorithms.

Introduction

Multi-document summarization is the process of generating a short version of given materials to indicate its main ideas. As the number of documents on the web exponentially increases, text summarization has attracted a growth of attention since it can help people get the topic within a short time.

Most existing studies are extraction-based methods. The extraction approaches usually use a rank model to select sentences from original text set. However, these methods suffer from severe problem that top-ranked sentences tend to convey much redundant information. Although some methods tried to reduce the redundancy (Li et al. 2009), finding balance between wide coverage and minimum redundancy is a non-trivial task.

In this paper, an ideal summary is assumed to represent the whole document set, namely, by reading the summarization instead of the whole set one can understand the general idea of the original documents. Rank models merely provide important sentences by score, and hence are not able to cover all aspects of the original corpus. Inspired by data compression and reconstruction(Simon, Snavely, and Seitz 2007a; Yang et al. 2013; Yu et al. 2014), a good summary should recover the whole documents, or in other words, reconstruct the whole documents. Based on the assumption, we think a good summary should meet three key requirements: **Coverage**, **Sparsity** and **Diversity**. **Coverage** means the extracted summary can conclude every aspect of all documents. Similar to (He et al. 2012), we use non-negative linear combination to represent the relations between sentences in the document set and the summary sentences. The relation can help to reconstruct the original document set by summary sentences. **Sparsity** means one certain sentence in the document set should be precisely represented by only a small number of summary sentences. Intuitively, multi-document set always have one central topic and some sub-topics, indicating that the summary sentences should also be categorized into groups. Each sentence in the document set should only be represented by summary sentences in the same group. On the contrary, not all summary sentences can be used to reconstruct one certain sentence although all of them are important. Otherwise we will bring in noise since some summary sentences are irrelevant. To enforce the sparsity property, we introduce sparse coding, a powerful tool, for denoising (Elad and Aharon 2006). **Diversity** means to eliminate redundancy. As stated above, document set can often be divided into some sub-topics, and thus we capture the overall view of the document set if we introduce diversity into our model. In this paper, we use the correlation of the least different summary sentence pairs to measure diversity.

Based on these three requisites, we design a two-level sparse representation model to tackle the multi-document summarization problem:

- *Level-1*: The summary set is a sparse representation of the original document set.
- *Level-2*: Each sentence in the candidate set is sparsely reconstructed by the summary set.

The model is illustrated in Figure 1. We denote the set of all the sentences in the original document set as candidate set, and the set of all the summary sentences as summary set. The summary set is selected from the candidate set. Similar to (Yang et al. 2013), the two-level sparse representation model we introduced is NP-hard, we use simulated annealing algorithm to get the summarization.

To our knowledge, this is the first paper that utilizes cover-

*Corresponding author

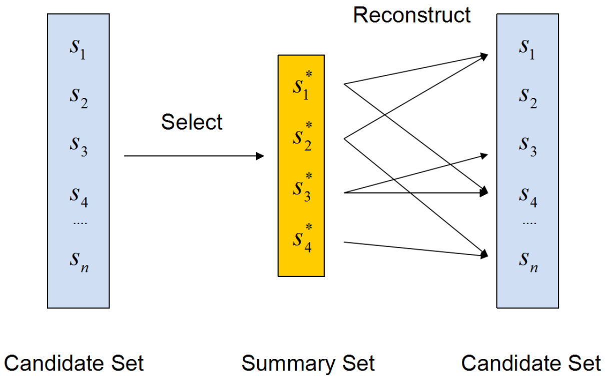

Figure 1: **two-level sparse representation model**

age, sparsity and diversity together in multi-document summarization. We test our model on DUC2006 and DUC2007 data sets and the results show that our approach can perform effectively and efficiently.

Related Work

Multi-document summarization aims at reducing the long documents into short length sentences, which helps readers quickly grasp the general information of the document set. Though over the past 50 years, the problem has been addressed from many different perspective in varying domains and using various paradigms, it is still a non-trivial task.

There are two main approaches in text summarization : text abstraction and text extraction.

Text abstraction build an internal semantic representation and then use natural language generation (Reiter, Dale, and Feng 2000) techniques to create a summary that is closer to what a human might generate. Such a summary might contain words not explicitly present in the original documents(Qian and Liu 2013). Though abstraction method uses smaller units such as words and phrases, it may contain more information than extraction method, but it suffers from poor readability and low efficiency, which is critical for online summarization.

Text extraction means to identify the most relevant sentences in one or more documents. Most existing systems use rank model to select the sentences with highest scores to form the summarization. They differ in the rules they used to compute the salient scores. The important parts are often retrieved by using some natural language processing and heuristics methods(Luhn 1958; Simon, Snavely, and Seitz 2007b). More advanced techniques consider the rhetorical structure(Marcu 1997) and semantic relationships(Gong and Liu 2001) and there are also some machine learning models(Kupiec, Pedersen, and Chen 1995). (Conroy and O'leary 2001) model the problem of extracting a sentence from a document using hidden Markov model(HMM). Graph based models like PageRank(Brin and Page 1998) and HITS(Kleinberg 1999) build similarity graph of sentences, and use influence propagation algorithms to give each sentence a score. One of the disadvantages in above techniques is that they seem to ignore the redundancy and coverage in summarization.

Sparse coding is proved to be very useful in image processing(denoting, in painting, super resolution) :(Yu, Sapiro, and Mallat 2012; Mairal, Elad, and Sapiro 2008; Yang et al. 2013) and object recognition:(Yang et al. 2009; Boureau et al. 2011). It was first introduced into document summarization in (He et al. 2012). They represent each sentence as a non-negative linear combination of the summary sentences. But they do not consider the sparsity of the summarization. We design a two-level sparse representation model to extract multi-document summarization. The original document set is sparsely represented by summary sentences. Each sentence in the original document set is sparsely represented by the summary sentences. In other words, we represent each sentence as a non-negative linear combination of only some of the summary sentences. Our experiments show that sparsity is critical for high quality summarizations.

Proposed Model

Preliminary

In this section, we give the notations we use in this paper. We denote the corpus of documents as $C_{corpus} = \{Doc_1, Doc_2, \dots\}$, in which Doc_i denotes the ith document in C_{corpus}. Each document is made up of a set of sentences, all of which form the **candidate set** $S = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$, where $\mathbf{s}_i \in R^d$ is the term-frequency vector for sentence i, and d is the number of distinct terms in the **candidate set**. The task of multi-document summarization is to select a small number of sentences $S^* = \{\mathbf{s}_1^*, \mathbf{s}_2^*, \dots, \mathbf{s}_k^*\}$ from the **candidate set** which best describe the subjects. Note that $k \ll n$ and $S^* \subset S$. Here, we denote S^* as **summary set**.

MDS-Sparse Model

In this section, we describe the details of our proposed framework MDS-Sparse (***M**ulti-**D**ocument **S**ummarization based on Two-Level **Sparse** Representation Model*).

The most challenging part is to properly model summaries. We observe that an effective multi-document summarization should meet three key requirements:

1. **Coverage.** Most existing extraction methods focus on selecting top ranked sentences, which are considered important in the whole document set. But in real life, a good summarization should not only contain the main ideas of the whole topic, it should conclude the whole document set. The summary sentences should represent the other sentences. Like (He et al. 2012), we find summary sentences that can best reconstruct other sentences by non-negative linear combination.

2. **Sparsity.** As multi-document texts often describe one central topic and some sub-topics, the sentences can be categorized into groups. Accordingly, we can only use summary sentences in the same group to conclude one certain sentence in the document set. Instead of using traditional classification methods, we use sparse coding technologies to impose the sparsity of the representation.

In fact, our model incorporate a two-level sparse representation model : (1) The **summary set** is the sparse representation of the **candidate set** (2) All sentences in the **candidate set** is sparsely represented by the **summary set**, which means only some of the sentences in the **summary set** is used when constructing one sentence in the **candidate set**.

3. **Diversity.** A topic which contains many documents usually contains one core topic and some subtopics. A good summarization should not only find the most obvious topic, but also other sub-topics that help us better understand the whole document set. Thus we use the correlation of the least different summary sentence pairs to eliminate redundancy and improve diversity.

Coverage By treating the problem of multi-document summarization as the issue of data reconstruction, each sentence in the original **candidate set** can be approximately reconstructed by a non-negative weighted linear combination of the **summary set**.

Given a sentence $\mathbf{s}_i \in S$, MDS-Sparse represents it as a non-negative linear combination of the **summary set**.

$$\begin{aligned} &\mathbf{s}_i \approx \sum_{j=1}^{k} a_{ji}\mathbf{s}_j^* \\ &s.t. \quad a_{ji} \geq 0, \end{aligned} \tag{1}$$

where $a_{ji} \geq 0$ is the coefficient of the linear combination. To be specific, a_{ji} is a measurement of the correlation between s_i and s_j^*. To evaluate the coverage of the summary, we define the reconstruction error in L$_2$-norm:

$$re(\mathbf{s}_i) = \|\mathbf{s}_i - \sum_{j=1}^{k} a_{ji}\mathbf{s}_j^*\|_2^2 \tag{2}$$

Then the loss function is the global reconstruction error of the summary set:

$$\begin{aligned} J &= \underset{\mathbf{S}^*,\mathbf{A}}{min} \sum_{i=1}^{n} re(\mathbf{s}_i) \\ &= \underset{\mathbf{S}^*,\mathbf{A}}{min} \sum_{i=1}^{n} \|\mathbf{s}_i - \sum_{j=1}^{k} a_{ji}\mathbf{s}_j^*\|_2^2 \\ &s.t. \quad a_{ji} \geq 0, \end{aligned} \tag{3}$$

where $\mathbf{S}^*$ is the matrix of the summary sentences and $\mathbf{A}$ is the matrix of a_{ji}.

Sparsity As stated before, our framework is a two-level sparse representation model. Each sentence in the **summary set** may contain different aspects of the core topic, and other sentences should be covered by only some of them. So we put sparsity restriction on $a_{:i}$, the columns of coefficient matrix, to ensure that each sentence is reconstructed by a small number of the summary sentences. Here we impose L$_0$-norm on each $a_{:i}$

$$\begin{aligned} &J = \underset{\mathbf{S}^*,\mathbf{A}}{min} \sum_{i=1}^{n} \|\mathbf{s}_i - \sum_{j=1}^{k} a_{ji}\mathbf{s}_j^*\|_2^2 + \lambda \sum_{i=1}^{n} \|a_{:i}\|_0 \\ &s.t. \quad a_{ji} \geq 0, \quad \lambda > 0, \end{aligned} \tag{4}$$

where L$_0$-norm controls the sparsity of $\mathbf{A}$ hence each sentence in the **candidate set** is represented by a small number of sentences in the **summary set**. Given the general intractability of the L$_0$-norm problem, we replace the L$_0$-norm with L$_1$-norm since L$_1$-norm problem is tractable and the two norms is known to yield similar results. The loss function therefore becomes:

$$\begin{aligned} &J = \underset{\mathbf{S}^*,\mathbf{A}}{min} \sum_{i=1}^{n} \|\mathbf{s}_i - \sum_{j=1}^{k} a_{ji}\mathbf{s}_j^*\|_2^2 + \lambda \sum_{i=1}^{n} \|a_{:i}\|_1 \\ &s.t. \quad a_{ji} \geq 0, \quad \lambda > 0 \end{aligned} \tag{5}$$

Diversity The summary sentences should be diverse, since we expect a summarization to involve many different subjects. In the objective function, we add the maximized correlation score rather than the average correlation score because the diversity of a summary is determined by the least different sentence pairs; while the mean value measurements do not guarantee that the member of any pair differs from each other to some degree(Yang et al. 2013). Our loss function now becomes:

$$\begin{aligned} J = \underset{\mathbf{S}^*,\mathbf{A}}{min} &\sum_{i=1}^{n} \|\mathbf{s}_i - \sum_{j=1}^{k} a_{ji}\mathbf{s}_j^*\|_2^2 \\ &+ \lambda \sum_{i=1}^{n} \|a_{:i}\|_1 + \beta max_{j \neq k} corr(\mathbf{s}_j^*, \mathbf{s}_k^*) \\ &s.t. \quad a_{ji} \geq 0, \quad \lambda > 0, \quad \beta > 0 \end{aligned} \tag{6}$$

The correlation function is defined below:

$$corr(\mathbf{s}_i^*, \mathbf{s}_j^*) = \frac{(\mathbf{s}_i^* - \overline{\mathbf{s}_i^*})(\mathbf{s}_j^* - \overline{\mathbf{s}_j^*})}{\sigma_i \sigma_j}, \tag{7}$$

where $\overline{\mathbf{s}^*}$ is the mean value of the vector and σ is the standard deviation.

Algorithm

The optimization problem defined in Eq.(6) is NP-hard(Yang et al. 2013). We use the simulated annealing algorithm to find the near optimal solution.

The overview of MDS-Sparse.

1. We set the initial temperature and create a random initial solution
2. The algorithm begins to loop until the stop criterion is met. Usually either the system has sufficiently cooled, or the system has not been promoted for certain number of iterations.
3. We fix the **summary set** to find a better coefficient matrix **A** that minimizes the reconstruction error.
4. From here we select a neighbor of each summary sentence by making a small change to our current **summary set**.

5. We then decide whether to move to that neighbor solution.
6. Finally, we decrease the temperature and continue looping (step to 2).

Algorithm 1 MDS-Sparse Algorithm

```
Input: candidate set $S$, the number of summary sentences $k$, sparse coefficient $\lambda$, correlation coefficient $\beta$, $k = 0$, $J$ is the loss function, $S^*$ is initialized randomly. $T_{stop}$ is the temperature that the annealing algorithm will stop.
Output: summary set $S^*$
1: while $T_k > T_{stop}$ do
2:     $A \leftarrow SparseCoding(S, S^*)$
3:     if $J(S, A, T_k) < J_{opti}$ then
4:         $J_{opti} \leftarrow J(S, A, S_k^*)$
5:         $S_{opti}^* \leftarrow S_k^*$
6:     else
7:         $rej \leftarrow rej + 1$
8:         if $ref \geq MaxConseRej$ then
9:             return $S_{opti}^*$
10:        end if
11:    end if
12:    for all $s^*$ in $S_k^*$ do
13:        $tmp \leftarrow Update\_S^*(s^*, T_k)$
14:        if Accept($s^*, tmp, S_k^*, T_k$) then
15:            $S_{k+1}^* \leftarrow S_{k+1}^* \cup tmp$
16:        else
17:            $S_{k+1}^* \leftarrow S_{k+1}^* \cup s^*$
18:        end if
19:    end for
20:    $T_{k+1} \leftarrow Update\_T(k)$
21:    $k \leftarrow k + 1$
22: end while
23: return $S_{opti}^*$
```

We first randomly select a temporary **summary set** and then use them to sparsely represent the original **candidate set**. We can fix the diversity part to constant since the temporary **summary set** is now fixed. Our loss function can be represented as:

$$\begin{aligned} J = \min_{\mathbf{S}^*, \mathbf{A}} & \sum_{i=1}^{n} \|\mathbf{s}_i - \sum_{j=1}^{k} a_{ji}\mathbf{s}_j^*\|_2^2 \\ & + \lambda \sum_{i=1}^{n} \|a_{:i}\|_1 + Constant \\ s.t. \quad & a_{ji} \geq 0, \quad \lambda > 0, \quad \beta > 0 \end{aligned} \tag{8}$$

The above convex optimization problem can be solved using multiplicative algorithm(Hoyer 2002). ,

$\cdot *$ and $\cdot /$ are element-wise multiplication and division respectively. A is calculated iteratively until convergence is met.

We randomly select neighbors from temporally selected sentences (temporary **summary set**). We iteratively update each sentence in this set by searching from its neighbors by using function **Update_S*** in line 13.

Algorithm 2 SparseCoding(S^*, S)

```
Input: candidate set $S$, summary set $S^*$
Output: coefficient matrix $A$
initialize $A$ with $\frac{1}{k}$
while $t < 100$ do
    $A^{t+1} = A^t \cdot *(S^{*T}S) \cdot /(S^{*T}S^*A^t + \lambda\mathbf{1})$
    if $norm(A^{t+1} - A^t) < 0.01$ then
        break
    end if
    $t \leftarrow t + 1$
end while
return $A^t$
```

The search range in **Update_S*** decreases monotonically as the temperature goes down. It shrinks rapidly at earlier iterations and decreases slowly in later steps, since as the iteration steps increase, the algorithm searches more locally to find the optimal solution.

The **Accept** function in line 14 is adopted to judge whether a replacement of the summary sentence is acceptable. Some replacements that fail to lower the loss function also have a chance to be accepted since they serve to allow exploring more of the possible space of solutions.

Experiments

Experimental Setup

Datasets In this study, we use the standard summarization benchmark DUC2006 and DUC2007 for evaluation. Document Understanding Conference (DUC) has organized yearly evaluation of document summarization. DUC2006 contains 50 document sets while DUC2007 contains 45 document sets. Every document set has 25 news articles. Each document set consists of several articles written by various authors, which is also the ground truth of the evaluation. Every sentence is either used in its entirety or not at all for constructing a summary. The length of a result summary is limited by 250 tokens(whitespace delimited).

Evaluation Metric We use the Rouge(Lin 2004) evaluation toolkit, which is adopted by DUC for automatic summarization evaluation. It measures summary quality by counting overlapping units such as the n-gram, word sequences and word pairs between the candidate summary and the reference summary. As mentioned in (He et al. 2012), Rouge-N is defined as follows:

$$\begin{aligned} & Rouge\!-\!N \\ = & \frac{\sum_{S \in Ref} \sum_{gram_n \in S} CountMatch(gram_n)}{\sum_{S \in Ref} \sum_{gram_n \in S} Count(gram_n)} \end{aligned} \tag{9}$$

where n stands for the length of the n-gram, Ref is the set of reference summaries. *CountMatch*$(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries, and *Count*$(gram_n)$ is the number of n-grams in the reference summaries. Among the evaluation methods implemented in Rouge, Rouge-1 focuses on the occurrence of the same words between candidate summary and reference summary, while Rouge-2 and

Rouge-SU4 focus more on the readability of the candidate summary. We use these three metrics in the experiment.

Compared methods We compare our MDS-Sparse with several state-of-the-art text extraction methods described briefly as follows:

1. **Random** selects sentences randomly from the **candidate set**.
2. **Lead** (Simon, Snavely, and Seitz 2007b) first sort all the documents chronologically and then select sentences from each document one by one.
3. **LSA** (Gong and Liu 2001) applies the singular value decomposition (SVD) on the terms-frequency matrix, then selects sentences with highest eigenvalues.
4. **DSDR** (He et al. 2012) represents each sentence as a non-negative linear combination of the summary sentences. And it uses sparse coding to select the summary sentences.

Preprocessing

In this subsection, we describe how we convert the raw documents to fit into our mathematical models.

Firstly we erase the html tag from the raw document. Secondly we segment the documents into sentences. The problem of sentence segmentation is non-trivial. In English and some other languages, using punctuation, particularly the full stop character is a reasonable approximation. However even in English this problem is not simple due to the use of the full stop character for abbreviations, which may or may not also terminate a sentence. For example, *Mr.* with a full stop character is not a sentence in "*Mr. Smith went to the shops.*" Fortunately, there are already mature tools we can use the tackle this challenge. We use splitta [1], which uses SVM as classifier to separate sentences. On splitta's homepage, the reported error rates on test news data are near 0.25%. After sentence segmentation, we eliminate the stop-words and use porter stemming algorithm [2] to stem each sentence. And finally, we create a term-frequency vector for every sentence. All the sentences form the **candidate set**.

Experimental Results

In this subsection, we give the results of the experiments and the analysis.

Overall performance Table 1 and Table 2 are the overall performance comparison of MDS-Sparse against other algorithms. Rouge generates three kinds of scores: precision, recall and F-measure. To compare different approaches, we employ F-measure, a widely used measurement which combines precision and recall, to fully disclose the performance of different algorithms.

MDS-Sparse+div denotes our algorithm incorporated with diversity component (We set $\beta = 1000$ through experiments). MDS-Sparse-div denotes our algorithm without diversity ($\beta = 0$). The scores in bold are the highest ones in the column. From Table 1 and Table 2, it is obvious that MDS-Sparse outperforms other algorithms significantly.

Algorithm	Rouge-1	Rouge-2	Rouge-SU4
Random	0.28047	0.04613	0.08785
Lead	0.30758	0.04836	0.08652
LSA	0.24415	0.03022	0.07097
DSDR	0.32034	0.04585	0.09804
MDS-Sparse+div	0.34034	**0.05233**	**0.10730**
MDS-Sparse-div	**0.34439**	0.05122	0.10717

Table 1: Average F-measure performance on DUC2006. MDS-Spsarse+div and MDS-Sparse-div denote MDS-Sparse with diversity and MDS-Sparse without diversity respectively.

Algorithm	Rouge-1	Rouge-2	Rouge-SU4
Random	0.30199	0.04628	0.08763
Lead	0.31188	0.0576	0.10201
LSA	0.25977	0.04062	0.08338
DSDR	0.32641	0.04876	0.10245
MDS-Sparse+div	0.35258	0.05479	0.11233
MDS-Sparse-div	**0.35399**	**0.06448**	**0.11669**

Table 2: Average F-measure performance on DUC2007. MDS-Sparse+div and MDS-Sparse-div denote MDS-Sparse with diversity and MDS-Sparse without diversity respectively.

Except MDS-Sparse, DSDR performs better than other algorithms, and it may be because DSDR also uses sparse coding. But it did not consider that one certain sentence should be sparsely represented by a small number of summary sentences, which is considered in our model. Besides, selecting the leading sentences (Lead) is a little better than just selecting sentences randomly. It may be caused by that article writers tend to put the conclusion sentences at the beginning of the document. Among all the six summarization algorithms, LSA shows the poorest performance on both data sets. LSA directly applies SVD on the term-frequency matrix and chooses those sentences with the largest indexes along the orthogonal latent semantic directions. Such sentences may be important of matrix decomposition, but it seems not helpful for human understanding.

Rouge-1 focuses on the occurrence of the same words between candidate summary and reference summary, while Rouge-2 and Rouge-SU4 focus more on the readability of the candidate summary. Since our work uses a two-level sparse representation model which is more close to real life, our model obtains highest scores in all three measurements. MDS-Sparse+div and MDS-Sparse-div act almost the same. This may be caused by that we adopt the non-negative linear combination of the summary sentences for reconstructing the candidate set. As mentioned in (He et al. 2012), the non-negative weighted combination might means addition of the summary sentences, and therefore, our framework incorporates diversity naturally.

Efficiency In addition, the speed of our methods is quite competitive. Note that we do not consider other algorithms, because both of MDS-Sparse and DSDR adopt sparse cod-

[1] https://code.google.com/p/splitta/

[2] http://tartarus.org/martin/PorterStemmer

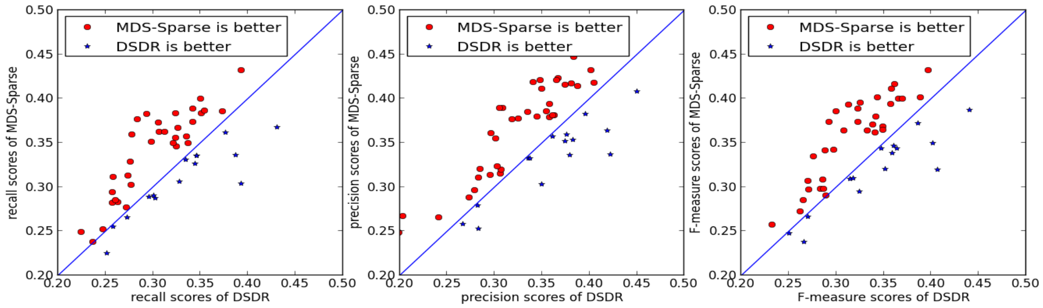

Figure 2: **The Rouge recall, precision, F-measure scores of MDS-Sparse and DSDR on each document set of DUC2006, the circles denote MDS-Sparse are better than DSDR, while the stars denote otherwise.**

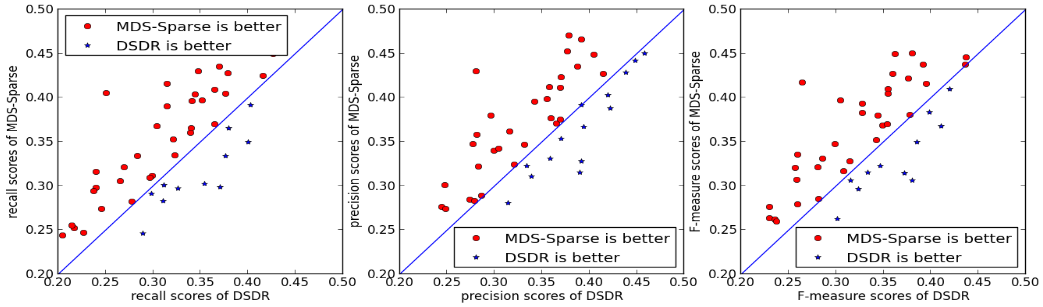

Figure 3: **The Rouge recall, precision, F-measure scores of MDS-Sparse and DSDR on each document set of DUC2007, the circles denote MDS-Sparse are better than DSDR, while the stars denote otherwise.**

ing approach. The experiments were performed on a 2.4GHz PC machine (Intel Core2 P8600) with 4GB of memory, running on an Ubuntu12.04 operating system. The average running time of DSDR on one document set is 4334.7s, while MDS-Sparse is 38.9s. Our algorithm runs two orders of magnitude faster than DSDR.

Evaluations on Different Topics In subsection , we discussed the overall performance of MDS-Sparse and other algorithms. In this subsection, we will focus on the topic-level comparison of the algorithms. We only compare MDS-Sparse+div ($\beta = 1000$ through experiments) and DSDR since both of them outperform other algorithms significantly and both use sparse coding framework to solve the summarization problem.

In Figure 2 and Figure 3, we compare the topic-level superiority of MDS-Sparse (We set $\beta = 1000$ through experiments) and DSDR in all the three measurements (recall, precision and F-measure). Each of the red circles and the blue stars denotes one document set which describes one topic in the document sets. The document set is a red circle when MDS-Sparse performs better than DSDR, otherwise a blue star. It is obvious that MDS-Sparse outperforms DSDR on both DUC2006 and DUC2007 data sets. The number of red circles is much more than that of blue stars.

Conclusion

In this paper, we propose a novel model to tackle the problem of multi-document summarization. We first investigate three requisite properties of an ideal summarization. Then a two-level sparse representation model is devised to extract all the salient sentences. In our model, the task of multi-document summarization is regarded as a document reconstruction problem which contains diversity naturally. Extensive experiments on standard datasets show that our methods is quite effective.

Recently a lot of new researches on text summarization like storyline generation and hierarchical summarization have attracted much attention. It would be of great interests to extend our model to tackle these relevant emerging tasks.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant No. 61170091).

References

Boureau, Y.-L.; Le Roux, N.; Bach, F.; Ponce, J.; and LeCun, Y. 2011. Ask the locals: multi-way local pooling for image recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2651–2658. IEEE.

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems* 30(1):107–117.

Conroy, J. M., and O'leary, D. P. 2001. Text summarization via hidden markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 406–407. ACM.

Elad, M., and Aharon, M. 2006. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on* 15(12):3736–3745.

Gong, Y., and Liu, X. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 19–25. ACM.

He, Z.; Chen, C.; Bu, J.; Wang, C.; Zhang, L.; Cai, D.; and He, X. 2012. Document summarization based on data reconstruction. In *AAAI*.

Hoyer, P. O. 2002. Non-negative sparse coding. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, 557–565. IEEE.

Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46(5):604–632.

Kupiec, J.; Pedersen, J.; and Chen, F. 1995. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, 68–73. ACM.

Li, L.; Zhou, K.; Xue, G.-R.; Zha, H.; and Yu, Y. 2009. Enhancing diversity, coverage and balance for summarization through structure learning. In *Proceedings of the 18th international conference on World wide web*, 71–80. ACM.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 74–81.

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development* 2(2):159–165.

Mairal, J.; Elad, M.; and Sapiro, G. 2008. Sparse representation for color image restoration. *Image Processing, IEEE Transactions on* 17(1):53–69.

Marcu, D. 1997. From discourse structures to text summaries. In *Proceedings of the ACL*, volume 97, 82–88. Citeseer.

Qian, X., and Liu, Y. 2013. Fast joint compression and summarization via graph cuts. In *EMNLP*, 1492–1502.

Reiter, E.; Dale, R.; and Feng, Z. 2000. *Building natural language generation systems*, volume 33. MIT Press.

Simon, I.; Snavely, N.; and Seitz, S. M. 2007a. Scene summarization for online image collections. In *ICCV*, volume 7, 1–8.

Simon, I.; Snavely, N.; and Seitz, S. M. 2007b. Scene summarization for online image collections. In *ICCV*, volume 7, 1–8.

Yang, J.; Yu, K.; Gong, Y.; and Huang, T. 2009. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 1794–1801. IEEE.

Yang, C.; Shen, J.; Peng, J.; and Fan, J. 2013. Image collection summarization via dictionary learning for sparse representation. *Pattern Recognition* 46(3):948–961.

Yu, H.; Deng, Z.-H.; Yang, Y.; and Xiong, T. 2014. A joint optimization model for image summarization based on image content and tags. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Yu, G.; Sapiro, G.; and Mallat, S. 2012. Solving inverse problems with piecewise linear estimators: from gaussian mixture models to structured sparsity. *Image Processing, IEEE Transactions on* 21(5):2481–2499.