Enhancing extractive summarization using non-negative matrix factorization with semantic aspects and sentence features

NGUYEN Thi Thu Trang
Hanoi University of Science and Technology
1 Dai Co Viet, Hai Ba Trung, Hanoi Vietnam
trangntt@soict.hust.edu.vn

LE Thanh Huong
Hanoi University of Science and Technology
1 Dai Co Viet, Hai Ba Trung, Hanoi Vietnam
huonglt@soict.hust.edu.vn

DUONG Viet Hung
Hanoi University of Science and Technology
1 Dai Co Viet, Hai Ba Trung, Hanoi Vietnam
hunglc007@gmail.com

ABSTRACT

The main task in extractive text summarization is to evaluate the important of sentences in a document. This paper aims at improving the quality of an unsupervised summarization method, i.e. non-negative matrix factorization, by using sentence features and considering semantically related words using word embeddings (i.e. word2vec) in sentence scoring. The experiments were carried out with different scenario using the DUC 2007 dataset. Experimental results showed that when NMF was combined three types of sentence features (i.e., surface, content, and relevant features) and word2vec, the system got best performance with 42.34% for Rouge-1 and 10.77% for Rouge-2, increasing 0.67% Rouge-1 and 0.78% Rouge-2 in compared with only NMF.

CCS CONCEPTS

• **Computing Methodologies → Artificial Intelligence**; *Information Extraction*

KEYWORDS

Text summarization, non-negative matrix, word embedding

ACM Reference Format:

NGUYEN Thi Thu Trang, LE Thanh Huong, DUONG Viet Hung. 2017. Enhancing extractive summarization using non-negative matrix factorization with semantic aspects and sentence features. In SoICT '17: Eighth International Symposium on Information and Communication Technology, December 7–8, 2017, Nha Trang City, Viet Nam. ACM, New York, NY, USA, 6 pages. https://doi.org/ 10.1145/3155133.3155188

1 INTRODUCTION

Automatic text summarization is the task of generating an abstract or a summary of a text. It gained widespread interest due to overwhelming amount of textual information available in electronic format. Text summarization techniques can be broadly grouped into extractive and abstractive summarization. Extractive approach selects the most salient sentences from the input document. Meanwhile, abstractive approach is relied on natural language processing techniques to paraphrase main contents of a document and maybe use extra linguistic information in order to generate the final summary. This approach is much more complex than extractive approach since it is difficult to generate summaries with good linguistic quality. Most research on text summarization is extractive-based [1][2][3] since it is more simple and robust for summarization of text.

Research in extractive summarization can be divided into rule-based approaches and machine learning ones. Rule-based approaches [4][5] often use statistical features such as cue phrases, keywords, position of sentence, proper noun, etc. to calculate score of sentences and select sentences with highest score to put in the summary. Some approaches use graph-based model [1][6] to represent relations among text elements (words or sentences). Sentences are weighted and then selected based on information about node connections in the graph.

Research using machine learning approaches can be divided into supervised learning and unsupervised ones. Supervised approaches [7][8] used large training corpus (i.e., human-generated summaries) to extract features of sentences that are put in summaries. Two problems with these approaches are domain-dependent and creating the training corpus is costly. Since it is hard to get such a large corpus, there is not much research following this approach.

Recently, many work on text summarization are based on unsupervised learning [9][10]. Instead of using a large training corpus, these approaches use some methods to discover relations among sentences inside a document. Sentences that most related with other sentences in the document are put in the summary.

In this paper, we propose a hybrid extractive single-document summarization system combining rule-based approach and unsupervised learning one. Semantic aspect is considered in unsupervised learning in order to improve the process of detecting relations among sentences.

The rest of this paper is organized as follows. Section 2 introduces related works to unsupervised learning approaches and features used in text summarization. Backgrounds of Non-negative Matrix Factorization (NMF) and the method of applying NMF to text summarization are introduced in Section 3. Section 4 and 5 analyze our approach to combine word2vec and sentence

SoICT '17, December 7–8, 2017, Nha Trang City, Viet Nam

ACM ISBN 978-1-4503-5328-1/17/12...$15.00
https://doi.org/ 10.1145/3155133.3155188

features to NMF, respectively. Our text summarization system is introduced in Section 6. Experimental results are given and discussed in Section 7. Finally, Section 8 concludes the paper and highlights some possible extensions of the work in this paper.

2 RELATED WORKS

The basic idea of text summarization using unsupervised learning is to compute relations among sentences. The research of [10] and [11] use Latent Semantic Analysis (LSA) to extract latent structures from a document. This method constructs a distributional semantics matrix that relates sentences and the terms they contain. A mathematical technique called singular value decomposition (SVD) is applied to the matrix to produce a set of concepts related to the sentence and terms. Their research shows that text summarization using LSA provides better results than keyword-based methods.

One problem is that its semantics matrix after applying SVD contains both negative and positive weighted terms. In other words, the weighted vector that represents semantic meaning of a sentence can have negative values. As a result, unimportant sentences may be put to the summary [12][13].

To deal with this problem, [9] used NMF instead of LSA to extract salient sentences. The non-negative constraints in NMF permit the system selecting more meaningful sentences for the summarization than those selected using LSA.

In single-document summarization, the NMF matrix represents relations between terms and sentences of a document. Each value in the NMF matrix is the count of a sentence's term in a document based on character matching, not semantic-matching. Because of that, semantic meaning is not considered thoroughly in NMF.

To deal with this problem, this paper proposes to integrate semantic-related words into NMF, in order to find terms similar to sentence's terms. The count of a sentence’s term is now calculated as the count of all terms referring to the same concept in that document. Another drawback of using NMF for text summarization is that it cannot take advantages of sentence features, which were used very effective in previous research on text summarization [4][5]. To investigate the important of different features in text summarization, [8] constructed a summarizing system using supervised learning. Four types of sentence features being considered in their experiments are surface, content, event and relevance ones. Features belong to each feature type are shown in Table 1.

Table 1: Sentence features for text summarization [8]

Feature types	*Feature*
Surface feature	Sentence position in the document
	The number of words in the sentence
	The number of quoted words in the sentence
Content feature	Centroid words
	Signature terms
	High frequency words
Event feature	Event term (verbs and action nouns)
	Event elements (one or more associated named entities).
Relevant feature	Similarity with the first sentence in the document
	Similarity with the first sentence in the paragraph
	PageRank value of the sentence based on the sentence map

Their experiments showed that the combination of surface features, content features, and relevant features provided best result in comparison with other set of features.

Based on the work of [8], the above three features are also combined with NMF in our system to score sentences. This process is described in Section 5. The next section presents our method of applying NMF for summarizing text.

3 NMF FOR TEXT NORMALIZATION

Non-negative Matrix Factorization – NMF [14] is a method to find latent structure from data. This method is often used to reduce the dimensionality of the data by combining attributes so that meaningful features are produced.

NMF factorizes a data matrix A into two matrices W and H such as all three matrices having no negative elements. NMF uses an iterative procedure to modify the initial values of W and H so that the product approaches A. The procedure terminates when the approximation error converges or the specified number of iterations is reached. More specifically, the approximation of A is $A \sim WH$ is achieved by minimizing the error function $|A\text{-}WH|$.

$$\min_{W \geq 0, H \geq 0} \left| A - WH \right|_F \qquad (1)$$

The method of calculating matrices W and H is described in details in the work of [9]. During model apply, an NMF model maps the original data into the new set of features discovered by the model. The number of new features is determined by users. A natural property of NMF is that it automatically clusters the columns of input data. Therefore, it is suitable for summarization problem.

In text summarization, a sentence is represented using a vector of words. A document is then a matrix that relates sentences and words of the document.

In our text summarization system, a document after removing stop words is represented by a set of sentences with each sentence is a set of terms. Matrix A is de-composited by the product of two smaller matrices W and H, as shown in **Figure 1**.

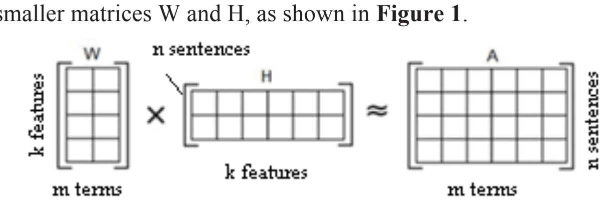

Figure 1: Non-negative Matrix Factorization for Text Summarization.

In Figure 1, A is an m x n matrix, W is an m x k matrix, and H is a k x n matrix. Here m is the number of document's terms, n is the number of sentences in the document, and k is the number of features to be produced. Each value in matrix A is the term-frequency of a term in a sentence. Generic Relevance of a Sentence (GRS) is then evaluated basing on matrix H as shown in Equation 2[9]. Sentences with the highest generic relevance values are put in the summary.

$$\text{GRS of a j}^{\text{th}}\text{ sentence} = \sum_{i=1}^{k} (H_{ij} * weight(H_{i*}) \quad (2)$$

where weight(H_{i*}) is calculated as follows:

$$weight(H_{i*}) = \frac{\sum_{q=1}^{n} H_{iq}}{\sum_{p=1}^{k} \sum_{q=1}^{n} H_{pq}} \quad (3)$$

weight(H_{i*}) is the relative relevance of the i'th feature among all features in matrix H. The generic relevance of a sentence refers to how much the sentence reflecting major topics, which are represented as k features in matrix H.

Steps to extract salient sentences from a document using NMF are as bellows:

Step 1. Preprocessing the input document by splitting it into sentences, perform stop-words removal and word-stemming operations.
Step 2. Transferring the document into a terms-by-sentences matrix (A).
Step 3. Decomposition matrix A into two sub matrices W and H
Step 4. Normalizing two matrices W and H
Step 5. Calculating generic relevance scores for each sentence using matrix H
Step 6. Select sentences with highest scores to put in the summary until the summary reaches the limitation length.

4 DEALING WITH SEMANTIC ASPECTS

In the extractive text summarization approach, the target is to select important concepts, phrases or sentences. The importance of those elements significantly depends on linguistic features of sentences. As mentioned in Section 2, we did combine the three structure features of sentences based on the work of [8] to improve the quality of NMF in text summarization: (i) surface features, (ii) content features, and (iii) relevant features.

In a normal extraction for these features, hidden links among words in the document are not considered. These links can exist in many different forms, such as cause-effect relationships, or emphasized words. In particular, there are a number of words that vary in presentation or expressing ways, but similar in meaning or context. We proposed that those semantic aspects should be covered in calculation of these sentence features for a better quality.

Traditional text processing treats words as discrete atomic symbols; therefore it cannot take advantages of the relationship between words. For example, if the system learns examples with the word "news", it cannot apply it to the word "article". Born from the idea that words appearing in the same contexts share the same meaning, words are embedded in a vector space where semantically similar words are located to nearby points.

In this paper, we propose to use word2vec model [15] to efficiently produce word embeddings from raw text, and hence to discover semantically related words. This model is a two-layer neural network that is trained to reconstruct linguistic contexts of words. It takes as input a large text corpus and generates a feature vector for each word in that corpus. These feature vectors are used to find semantically related terms using cosine measure.

We did normalization all the feature scores so that we can easily combine them to NMF. To measure the similarity of two sentences in Content and Relevant features, we propose to use doc2vec, which added another feature vector (i.e. document unique) to get a numeric representation of the document.

4.1 Surface Features

Surface features are decided based on structure of documents or sentences, including (i) position feature and (ii) length feature. The surface score is the total of position score and length score.

i) Position Feature

Sentences in the beginning of the document normally contain abstractions or main topics. In our work, the first sentence in the document is the most important sentence, and so on. For a given sentence S_i (where i is the position of the sentence from the beginning of the document) the length score is calculated as the Equation 4.

$$PositionScore(S_i) = \frac{1}{i} \quad (4)$$

ii) Length Feature

The length feature is extracted based on the number of words in the sentence. Sentences with smaller size seems not containing important content due to the average of sentence size (i.e. measured by number of words, except stop words) tends to be stable in a specific domain (i.e. 10 for our experiment data). For a given sentence S_i, the length score is calculated as the Equation 5.

$$LengthScore(S_i) = \begin{cases} 0 & if\ length(S_i) \le 10 \\ \frac{length(S_i) - 10}{10} & if\ length(S_i) > 10 \end{cases} \quad (5)$$

4.2 Content Features

In this feature type, we investigated two well-known sentence features based on content-bearing words i.e., centroid words, and high frequency words, for both unigram and bigram.

i) Centroid Feature

For the centroid word feature, the sum of the weights of centroid words was calculated for each sentence. The score of centroid feature is calculated as the Equation 6.

$$CentroidScore(S_i) = \sum_{k=0}^{n} Similarity(S_i, S_k) \quad (6)$$

where:

- S_i: The i^{th} sentence to be calculated
- n: The number of sentences in the document
- $Similarity(S_i, S_k)$: The similarity between the sentence S_i and S_k

The two sentences are vectorized by two approaches: (i) using word frequencies in the sentence, and (ii) using doc2vec. The similarity between two sentences is finally measured by the cosine similarity of their vectors.

ii) High Frequency Word Feature

The sum of the weights of high frequency words is calculated for each sentence. The weight of the word w_k is computed by the Equation 7.

$$HighFrequencyWordScore\ (S_i) = \sum_{k=1}^{n} \frac{count(w_k)}{M} \quad (7)$$

where:

- n: number of high frequency words in the sentence S_i
- count(w_k): the number of appearance of the word w_k in the document.
- M: the number of words in the document

In order to cover word relationships in context, word frequencies in combination of the frequency of similar words, as illustrated in Equation 8. Similar terms are found using a threshold for word similarity using word2vec.

$$frequency(w_i) = count(w_i) + count(similar\_words(w_i)) \quad (8)$$

where:

- count(w_i) is the number of appearance of the word w_i in the document.
- count(similar_words(w_i)) is the number of appearance of similar words to the word w_i in the document.

4.3 Relevant Features

Relevant features are used to exploit inter-sentence relationships. We can assume that sentences related to important sentences or too many other sentences are important. At the beginning, first sentences of paragraphs are important. The relationship between a sentence (from the second sentence) and the first sentence is calculated. Relevant score is total of TextRank score and the relevance of each sentence to the first sentence in a paragraph. We adopted the PageRank algorithm to compute the TextRank score. The relevance of two sentences is calculated the same as the Centroid feature.

5 SYSTEM ARCHITECTURE

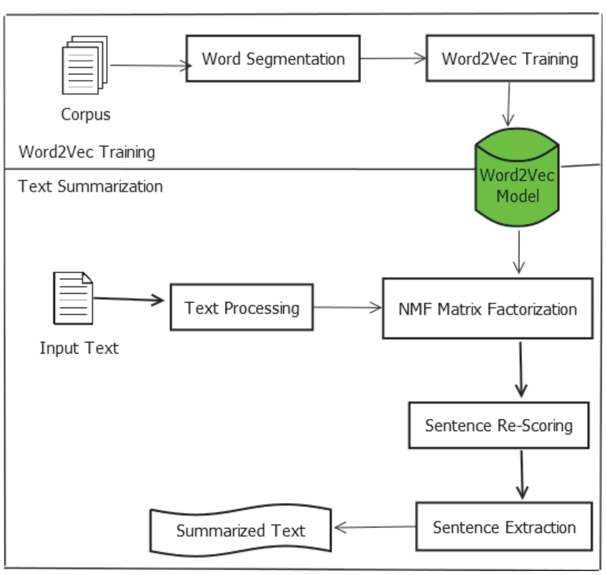

Figure 2: System architecture of text summarization.

The proposed architecture of our text summarization system is represented in **Figure 2**. The word2vec model was built to provide word similarity information for the NMF process. In the summarization phase, firstly, the input text is tokenized, normalized and POS tagged. Secondly, NMF matrix is factorized with the concern of similar terms using the word2vec model. The three feature scores are finally combined in re-scoring the sentences. Sentences are ranked and extracted by these final scores until the summarized text reaches the desired length.

5.1 Word2Vec Training Part

A huge corpus of text was provided to train for a word2vec model. Documents in this corpus are segmented into sentences, and then into words. After the training process, a word2vec model is created and stored for the text summarization part.

5.2 Text Summarization Part

Different from the Word2Vec Training part, the text processing in this part includes sentence detection, word segmentation, stopword removal, word normalization, lemmatizing, stemming, and POS tagger.

In the NMF Matrix Factorization task, a terms-by-sentences matrix is transferred from the input text. The matrix A is then decomposed into two sub non-negative matrices, i.e. a sematic feature matrix W and a semantic variable matrix H. Generic relevance score (GRS) for each sentence is calculated from the matrix H.

In the Sentence Re-scoring task, the final scores of sentences are computed by the total of GRS from NMF, and all the sentence feature scores (with hidden semantic using word2vec).

Eventually, most important sentences are extracted using final scores until the summary reaches the limitation length.

6 EXPERIMENT

6.1 Datasets

i) Word2Vec Training

The Continuous Skip-Gram of Gensim was adopted in our experiment for word2vec training. Based on empirical evidence, it was finally decided for the vector dimension of 300, the context window size of 5 preceding words and 5 succeeding words.

In order to get good sets of word embeddings, Gensim needs a large set of free text to be used as the system's input. Two free text corpus were combined in our system to form such a large corpus. They are: (i) Open American National Corpus (OANC) and (ii) English Wikipedia Corpus.

OANC is a big American English corpus (since 1990) including different types such as novel, newspaper, and dialogs. This corpus consists of 15 billions words in American English with automatically produced annotations for a variety of linguistic phenomena. English Wikipedia Corpus is a largest set of text taken from Wikipedia in English, whose size is approximately 12GB.

ii) Test Data – DUC 2007

The corpus of the Document Understanding Conference (DUC) was used to evaluate our text summarization system. However, there are test data for single document summarization only in Task 1 of DUC 2004 [16]. This provides very short summaries (i.e. headlines) from 500 newspaper and newswire articles. The maximum target length for very short summaries was 75 bytes. Therefore, previous works had to truncate summaries longer than 75 bytes before evaluation without bonus for creating summaries less than the target length (the space is there to be used). Since the summaries were too short, the evaluation result is not good, and semantic aspects cannot be enough captured.

In the DUC 2007, we observed that in a cluster, the content of documents is not duplicated despite of the same topic. Therefore, to build the test data, we concatenate documents in a cluster to a new long document, and the summary of each cluster now is the summary of that new document. We have finally 50 documents with 50 summaries of 250 words.

6.2 Experiment Results

Two kinds of experiments were carried out with our text summarizing system. The first one was conducted to find the optimal feature set, whereas the second one was to evaluate the system performance with or without using word2vec and sentence features.

Table 2: Rouge scores of the summarization system using different feature sets

Feature \ Rouge	Rouge-1 (%)	Rouge-2 (%)
Relevance	41.92	10.48
Relevance + W2V	42.35	10.53
Surface	39.04	9.29
Content	42.37	10.69
Content + W2V	41.36	9.73
NMF + Relevance + W2V	**42.62**	**10.63**
NMF + Surface + W2V	40.78	10.44
NMF + Content + W2V	42.30	10.65
NMF + Surface + Relevance + W2V	40.84	10.39
NMF + Relevance + Content + W2V	41.40	10.23
NMF + Surface + Content + W2V	41.32	10.34
NMF + 3 Features + W2V	**42.34**	**10.77**

In the first experiment, a baseline text summarization system using NMF was developed. Each feature type (i.e., relevant features, surface features, content features) was tested separately or in combination with NMF, word2vec and other feature types in order to find the optimal feature set. Our experiment results were reported in Table 3 and Figure 3, using Rouge-1 and Rouge-2 measures. ROUGE 2.0 (2017) - a Java package for evaluation of summarization tasks - was used in our experiments to calculate these measures.

Experiments with a single feature showed that the content feature was the best one among three feature types with 42.37% for Rouge-1 and 10.69% for Rouge-2. This confirmed that Content feature was the most important one for text summarization task. Relevant feature was the second important one with slightly smaller scores (41.92% and 10.48%).

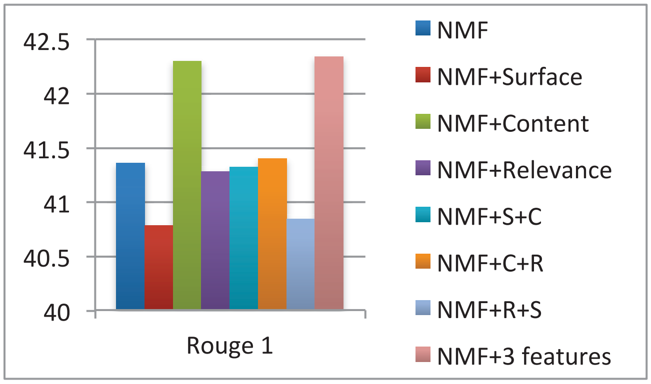

Figure 3: Rouge scores of the summarization system using different feature sets.

When all the three feature types were integrated with NMF and word2vec, the system achieved the highest Rouge-2 score (10.77%). Meanwhile, the system got the highest Rouge-1 score (42.62%) when relevance features were combined with word2vec and NMF.

In our second experiments, to evaluate the effect of word2vec and three features to the system performance, our text summarization system was developed with different strategies: a system using merely NMF; a system using three features; a system using three features and word2vec; a system using NMF and three features; a system using NMF, three features and word2vec.

Table 3: Rouge scores of the summarization system using different strategies

Strategy \ Rouge	Rouge-1 (%)	Rouge-2 (%)
NMF	41.67	9.99
3 features	41.56	10.42
3 features + W2V	41.98	10.44
NMF + 3 features	41.92	10.98
NMF + 3 features + W2V	**42.34**	**10.77**

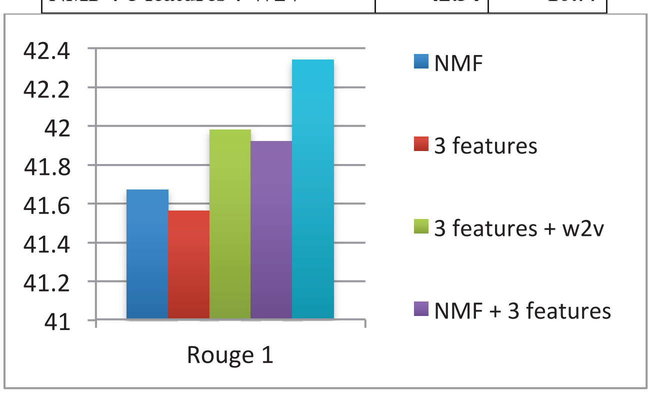

Figure 4: Rouge scores of the summarization system using different strategies.

Experimental results in Table 2 and Table 3 shown that integrating word2vec with the system did not always provide a better result. Combining three features was not also a good choice since it reduced the system performance in compared with the case of using each feature separately. However, combining three features with word2vec and NMF increases 0.78% for Rouge-1 and 0.35% for Rouge-2 in compared with the case of using only three features. This combination also provides better results, compared to the basic NMF (increasing 0.67% Rouge-1 and 0.78% Rouge-2).

7 CONCLUSIONS

This paper has proposed an approach to extractive text summarization using non-negative matrix factorization, in combination with word2vec and sentence features. Our experiments were carried out with different scenario using DUC 2007 dataset. Experimental results showed that when NMF was combined with three types of sentence features (i.e., surface, content, and relevant features) and word2vec, the Rouge-1 and Rouge-2 measures of the system increase 0.67% and 0.78%, respectively, in compared with the basic NMF.

In the future, we plan to investigate a method to optimize weights for each combination: combination among three feature types, and combination between GRS and feature scores. In addition, at the moment, Rouge measures are based on character matching, thus two sentences that are similar in meaning may be considered as different. Another possible work is to improve Rouge measures so that these measures can deal with semantic aspects.

REFERENCES

[1] G. Erkan and D. R. Radev, "LexRank: graph-based lexical centrality as salience in text summarization," *J. Artif. Intell. Res.*, vol. 22, no. 1, pp. 457–479, Jul. 2004.

[2] W. Li, M. Wu, Q. Lu, W. Xu, and C. Yuan, "Extractive Summarization Using Inter- and Intra- Event Relevance," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA, 2006, pp. 369–376.

[3] C.-Y. Lin and E. Hovy, "The Automated Acquisition of Topic Signatures for Text Summarization," in *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, Stroudsburg, PA, USA, 2000, pp. 495–501.

[4] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," 1998, pp. 335–336.

[5] H. P. Edmundson, "New Methods in Automatic Extracting," *J. ACM*, vol. 16, no. 2, pp. 264–285, Apr. 1969.

[6] M. Rada and P. Tarau, "TextRank: Bringing Order into Text," in *EMNLP*, 2004, vol. 4, pp. 404–411.

[7] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen, "Document Summarization Using Conditional Random Fields," in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, San Francisco, CA, USA, 2007, pp. 2862–2867.

[8] K.-F. Wong, M. Wu, and W. Li, "Extractive Summarization Using Supervised and Semi-supervised Learning," in *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, Stroudsburg, PA, USA, 2008, pp. 985–992.

[9] J.-H. Lee, S. Park, C.-M. Ahn, and D. Kim, "Automatic generic document summarization based on non-negative matrix factorization," *Inf. Process. Manag.*, vol. 45, no. 1, pp. 20–34, Jan. 2009.

[10] J.-Y. Yeh, H.-R. Ke, W.-P. Yang, and I.-H. Meng, "Text summarization using a trainable summarizer and latent semantic analysis," *Inf. Process. Manag.*, vol. 41, no. 1, pp. 75–95, Jan. 2005.

[11] Y. Wang and J. Ma, "A Comprehensive Method for Text Summarization Based on Latent Semantic Analysis," in *Natural Language Processing and Chinese Computing*, Springer, Berlin, Heidelberg, 2013, pp. 394–401.

[12] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[13] H. Zha, "Generic Summarization and Keyphrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2002, pp. 113–120.

[14] C.-J. Lin, "Projected Gradient Methods for Nonnegative Matrix Factorization," *Neural Comput*, vol. 19, no. 10, pp. 2756–2779, Oct. 2007.

[15] R. Řehůřek and P. Sojka, *Software Framework for Topic Modelling with Large Corpora*. University of Malta, 2010.

[16] DUC2004, "DUC 2004 corpus for text summarization." [Online]. Available: http://duc.nist.gov/duc2004/.