Social Context Summarization with Matrix Co-factorization[☆]

Minh-Tien Nguyen[a,b,*], Tran Viet Cuong[c], Nguyen Xuan Hoai[d], Le-Minh Nguyen[b]

[a]*Faculty of Information Technology,*
Hung Yen University of Technology and Education, Hung Yen, Vietnam.
[b]*School of Information Science,*
Japan Advanced Institute of Science and Technology (JAIST),
1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan.
[c]*Hanoi University of Science and Technology, Hanoi, Vietnam.*
[d]*AI Academy Vietnam and R&D Department, Anvita Jsc., Hanoi, Vietnam.*

Abstract

In the context of social media, users usually post relevant information corresponding to the contents of events mentioned in a Web document. This information (called by users posts) posses two important values in that (i) it reflects the content of an event and (ii) it shares hidden topics with sentences in the main document. In this paper, we present a novel model to capture the nature of relationships between sentences and user posts in sharing hidden topics for summarization. Unlike previous methods which are usually based on handcrafted features, our approach ranks sentences and user posts based on their importance to the topics. The sentence-user-post relation is formulated in a share topic matrix, which presents their mutual reinforcement support. Our proposed matrix co-factorization algorithm computes the score of each sentence and user post and extracts the top m ranked sentences and m user posts as

[☆] This manuscript is an improved and extended version of the awarded best paper: Utilizing User Posts to Enrich Web Document Summarization with Matrix Co-factorization, presented at *The Eighth International Symposium on Information and Communication Technology* (SoICT) 2017, Nha Trang, Vietnam. This work was done when the author studying PhD at JAIST, Japan.

[*] Corresponding author: Minh-Tien Nguyen

Email addresses: tiennm@utehy.edu.vn (Minh-Tien Nguyen), cuongtii100@gmail.com (Tran Viet Cuong), nxhoai@anvita.com.vn (Nguyen Xuan Hoai), nguyenml@jaist.ac.jp (Le-Minh Nguyen)

a summary. Experimental results on three datasets in two languages, English and Vietnamese, of social context summarization and DUC 2004 confirm the efficiency of our model in summarizing Web documents.

Keywords: Data Mining, Information Retrieval, Document Summarization, Social Context Summarization, Matrix Factorization.

1. Introduction

Text summarization is a challenging task which extracts salient information of a document while preserving its important meaning (Nenkova and McKeown, 2011). There are two major approaches for summarization: extraction and abstraction. The former tries to extract a small number of sentences from a document to form the summary whereas the later generates the summary which is close to the writing style of humans by considering several aspects such as readability and grammaticality. Outputs of a summarization system benefits many applications such as Web search or highlight generation. For example, search engines likes Google or Bing usually provide short pieces of information for Web documents or news providers, e.g. Yahoo News shows short sentences to reveal the main contents of a news article. Such beneficial usage demands high-quality text summarization systems.

In the context of social media, users have an ability to discuss events and topics mentioned in a Web document by posting their messages. For instance, Yahoo News provides both news articles and a Web interface where readers can write their comments on an event such as `"Germanwings crash families could seek damages in the U.S.: lawyer"`. These comments (user posts), also called social information (Yang et al., 2011; Wei and Gao, 2014; Nguyen and Nguyen, 2016), reflect the content of the news article by sharing common topics denoted in form of common words or phrases. Table 1[1] shows an example,

[1] https://www.yahoo.com/news/germanwings-crash-families-could-seek-damages-u-lawyer-140040694–sector.html

indicating that (i) the sentence is important because it includes important information for inferring the event. (ii) The comment also contains salient words or phrases, which form common hidden topics between the sentence and comment. (iii) Several well-written comments can be directly used as a summary, which enrich information in main sentences. This raises a challenging task of how to exploit relevant user posts to produce high-quality summaries.

Table 1: An extraction example.

[S]: **Families** of the **victims** of the **Germanwings** crash are considering filing a **claim for damages** in the United States if they cannot reach agreement with parent airline Lufthansa in Germany, a lawyer representing the families said on Sunday.
[C]: I'm sad for the people who are no longer on their life of this **Germanwings** strategy, but I don't know how much **money relatives** want for the flesh of **who die** on **Germanwings** case.

In this paper, we propose a novel summarization approach based on non-negative matrix co-factorization to automatically extract summary sentences and user posts of a Web document by incorporating its social context. The motivation of our model comes from the fact that sentences and user posts share common hidden topics in term of common words or phrases. For example, in Table 1, an extracted sentence and comment share common or inferred words in bold such as *"victims"* $\sim$ *"who die"*, *"Germanwings"*, *"families"* $\sim$ *"relatives"*. These terms form the hidden topics presenting the nature of relationships between sentences and comments. With this observation, we represent sentences and user posts in a share topic-matrix, which formulates the common topics in a mutual information fashion. Summaries are extracted by analyzing the matrix using a non-negative matrix co-factorization approach. The main contributions of the paper are as follow:

- Our new approach simulates the nature of mutual relationships between sentences and user posts in sharing hidden topics represented by a share topic-matrix. The matrix is found and used by matrix co-factorization to find salient sentences and user posts. To the best of our knowledge, we are the first to use matrix co-factorization for this summarization task.

- It investigates a normalization mechanism to avoid over-fitting during optimization. It also observes several aspects of our model such as the number of topics and the influence of data size of user posts, which benefit the literature.
- It conducts extensive experiments on three datasets of the social context summarization task and one benchmark dataset of text summarization. ROUGE-scores validate the efficiency of our model.
- It presents a robust model for summarizing Web documents, which include user posts. Our model is unsupervised suggesting that it can be applied to unrestricted domains. Our code is publicly accessible.[2]

We apply our model to the task of sentence and highlight extraction on three datasets in two languages, English and Vietnamese, and also on DUC 2004. ROUGE-scores confirm the efficiency of our model compared to the state-of-the-art methods for text and social context summarization. The rest of the paper is organized as follows. In section 2, we give a review of the related works. The notations are introduced in Section 3. We detail our framework in Section 4 and show the statistical analysis in Section 5. After extracting summaries, we show experimental results along with discussion and analyses in Section 6. The paper concludes with section 7.

2. Related Work

2.1. Extractive Summarization

Text summarization is a challenging task which has a long history dated back to 1950s. Luhn, 1958 investigated significant components of sentences such as high-frequency content words and sentence positions to summarize documents.

[2] https://drive.google.com/file/d/0ByufX58sGMBIaElfcGREOFVMZU0/view?usp=sharing

Recently, with the success of machine learning, it has been widely applied to text summarization (Yeh et al., 2005; Shen et al., 2007). The key idea of these studies is to formulate summarization as a binary classification problem and train a classifier to distinguish summary (label 1) or non-summary sentences (label 0). For instance, Yeh et al., 2005 combined classification and latent semantic analysis (LSA) to build summarizers. The score of a sentence is ranked by using features or a semantic matrix created by LSA. The highest F-score is 0.49 with 30% compression. Shen et al., 2007 exploited the sequence aspect of sentences to train a sequence labeling classifier to extract summaries. This method achieves 0.483 of ROUGE-2[3] and 0.419 of F-1 on DUC 2001.

Using submodular functions for document summarization also obtains promising results (Lin and Bilmes, 2011). Each function in their model guarantees the representativeness and diversity of a summary. This method obtains the best results of recall and F-score over DUC 2004 to 2007. Another direction is to formulate extraction in the form of concepts solved by Integer Linear Programming (ILP) (Woodsend and Lapata, 2010). The authors represent concepts as phrases extracted from a dependency tree then use ILP to acquire global optimization. This method achieves 0.25 for both ROUGE-1 and F-score on their datasets. There are also many studies of extractive summarization such as LexRank (Erkan and Radev, 2004), TextRank (Mihalcea and Tarau, 2004), or deep learning (Cao et al., 2015; Nallapati et al., 2016; Ren et al., 2017).

Matrix factorization has also been applied to summarization (Wang et al., 2008; Lee et al., 2009; Park et al., 2006; Nguyen et al., 2017a). Wang et al., 2008 presented a multi-document summarization model which analyzes sentence-level semantic matrix by using symmetric non-negative matrix factorization (NMF). A summary is formed by selecting the most informative sentences in each cluster. Lee et al., 2009 analyzed NMF and LSA and pointed out that NMF output better results which close to the human cognition process. In this work, we extend NMF for our task by exploiting relevant user posts for Web document

[3] https://en.wikipedia.org/wiki/ROUGE_(metric)

summarization. However. our model is different from Wang et al., 2008 and Lee et al., 2009 in that it integrates user posts into the summarization process and presents a matrix co-factorization algorithm for jointly extracting summaries.

2.2. Social Context Summarization

Amitay and Paris, 2000 were perhaps the first study on exploiting hyperlinks of a Web document for summarization. The authors assumed that a linked document reflects the content of an original document. The authors built InCommonSense, including hypertext retrieval and description selection (using classification) to extract a paragraph from the linked document as a summary. The average of user voting (1 to 5) is 4.71 compared to 4.14 of AltaVisata-style and 4.13 of Google-style. Delort et al., 2003 extended the hyperlink assumption by considering whole linked documents as the context instead of using paragraphs including hyperlinks as (Amitay and Paris, 2000). The authors propose content and context summarization algorithms based on similarity measurements. The best result is 0.45 in term of similarity in the content and context.

In Sun et al., 2005, the authors utilized click-through data retrieved from search engines to extract salient sentences in a Web document. The assumption of this method is that query keywords from users typed on search engines usually reflect the content of a Web document. Based on that, they present two summarization algorithms using an adaptation of significant words (Luhn, 1958) and LSA (Gong and Liu, 2001). ROUGE-1 is around 0.55 and 0.20 on their datasets. This method, however, has an issue that pages which users click on may be irrelevant to their interests.

Delort, 2006 extracted summaries by using the link analysis from sentences to comments represented by clusters. This method acquires 50% extracted matches corresponding to post summaries. Hu et al., 2008 selected sentences that best represent topics discussed among readers in blog posts. This approach first derives salient words denoted in three graphs: topic, quotation, and mention from comments. Summary sentences are next extracted by calculating the distance from each sentence to the graphs. The method obtains 0.64 of ROUGE-1 and

0.60 of NDCG[4] on their datasets.

Tweets from Twitter have widely been used to support sentences in selecting summaries. Yang et al., 2011 presented a dual wing factor graph model which uses classification as a preliminary step for incorporating tweets into the summarization process. The authors use a ranking method, which approximates an objective function to select both summary sentences and tweets. The performance of the model is 0.615 of ROUGE-1 and 0.500 of ROUGE-2 on five collected datasets. Gao et al., 2012 introduced a cross-collection topic-aspect model (cc-TAM) as a preliminary step to generate a bipartite graph used by co-ranking to select sentences and tweets for multi-document summarization. ROUGE-1 is 0.55 for document and 0.67 for tweet selection. Wei and Gao, 2014 integrated human knowledge into the summarization process by proposing local and cross features used for a learning-to-rank model in a news highlight extraction task. The model selects top m ranked sentences and tweets as a summary. ROUGE-1 is 0.292 and 0.295 for document and tweet extraction, respectively. Wei and Gao, 2015 extended LexRank by using auxiliary tweets for building a heterogeneous graph random walk (HGRW) to summarize single documents. It achieves 0.298 of ROUGE-1 when combining extracted sentences and tweets on their dataset. Li et al., 2016 introduced an ILP-based extraction model by exploring relevant Facebook public posts for extracting summaries. The authors defined different methods to embed information in public posts to estimate bi-grams weights used by ILP models with promising results.

The SoRTESum system (Nguyen and Nguyen, 2016; Nguyen and Nguyen, 2017) is perhaps the most relevant to our study. It models sentence-tweet relationships by using a set of lexical-level features to score sentences and tweets. The score of a sentence or tweet is computed in a reinforcement fashion. Finally, it extracts top m sentences and tweets, which have the highest scores. However, these sentence-tweet features are usually hand-crafted, domain-dependent, and difficult to represent the nature of relationships between sentences and user

[4] https://en.wikipedia.org/wiki/Discounted_cumulative_gain

posts. Our novel model in this paper differs from Nguyen and Nguyen, 2016 and Nguyen and Nguyen, 2017 in that it exploits the share topics between sentences and user posts to rank them by using non-negative matrix co-factorization instead of using feature-driven ranking, which is limited with domain adaptation.

3. Definitions

User posts. We define user posts as relevant comments or tweets generated from readers after reading a Web document. Formally, given a document d and a set of users $U = \{u_1, ..., u_n\}$, who read d, relevant user posts of d are denoted by UP_d created by $d \xrightarrow[U]{posting} C$ or $d \xrightarrow[U]{posting} T$, where C or T is a set of comments or tweets, $\xrightarrow[U]{posting}$ presents that C or T is produced by U.

Social context. There are several ways to formulate social context; however, we follow Yang et al., 2011 to define the social context of a Web document d by C_d presented by $\langle S_d, UP_d, U_d \rangle$, where S_d is a set of sentences in document d, UP_d is a set of relevant tweets or comments of d written by users U_d. In this study, we eliminate the user aspect because it is an implicit factor.

Summarization. The summarization task is to select important sentences and representative user posts as a summary by using C_d given the original document d. Representative user posts are those that also reflect the most important contents of a document. The intuition of this extraction is that while extracted sentences include salient information, user posts can also provide new information from users, which may be unavailable in main documents. The definition of the task indicates that the importance estimation is now for both sentences and user posts, instead of only for sentences as traditional methods.

4. Summarization with Matrix Co-factorization

Before introducing our model, we first show a general framework of summarization by using NMF. In Figure 1, the framework includes three major steps: document representation, sentence scoring, and sentence selection. The system receives a document as an input and presents a document as a matrix, which is usually in the form of a term-sentence matrix. The scoring step analyzes the matrix to create its weights, which are used to rank sentences. The selection step extracts important sentences based on their weights.

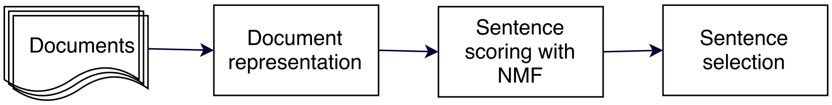


Figure 1: A general framework of summarization with matrix factorization.

In next sections, we first describe data preparation (the input) and the representation step. Then, we show a basic model based on NMF and introduce our model, which integrates social information into the scoring step. We finally show the selection step to extract summaries.

4.1. Data Preparation

Because DUC datasets lack social information, we, therefore, prepared thee datasets in two languages, English and Vietnamese, for our task.

SoLSCSum. This is a dataset created for social context summarization (Nguyen et al., 2016c). It includes 157 open-domain articles with 3,462 sentences, 25,633 comments collected from Yahoo News. Each sentence or comment was assigned a label by humans indicating whether a sentence is important or not. References were created by selecting both important sentences and comments, resulting 5,858 in total. Cohen's Kappa is 0.5845 with 95% confidence.

USAToday-CNN. This is a dataset for news highlight extraction (Nguyen et al., 2017b) derived from Wei and Gao, 2014. It includes 121 events retrieved from USAToday and CNN. Tweets were crawled from Twitter. After removing near-duplicate tweets, the authors asked annotators to give a label for each sentence and tweet. Cohen's Kappa is 0.6170 with 95% confidence.

VSoLSCSum. To validate our model in a non-English language, we used a Vietnamese dataset created for this task (Nguyen et al., 2016a). It includes 141 news articles and their relevant comments collected from several Vietnamese web pages. Annotators involved to create a label for each sentence and comment (summary or non-summary). Cohen's Kappa is 0.6850 with 95% confidence.

Table 2: The statistics of three datasets.

Dataset	#doc	#sents	#posts	#refs	ref type	language	label
SoLSCSum	157	3,462	25,633	5,858	extraction	English	Yes
USAToday-CNN	121	6,413	78,419	455	highlight	English	Yes
VSoLSCSum	141	3,760	6,926	2,448	extraction	Vietnamese	Yes

From Table 2, we can observe that the number of user posts is large enough to support sentences. In addition, the reference type of SoLSCSum and VSoLSCSum indicates that their references are created by selecting both important sentences and comments whereas the type of USAToday-CNN shows that its references are highlights created by humans in an abstractive way.

Table 3: Statistical observation; s: sentences, c: comments, and t: tweets.

Dataset	Observation	sentences (%)	user posts (%)
SoLSCSum	% overlapping	s/c: 13.26	c/s: 42.05
	% overlapping (no stop words)	s/c: 8.90	c/s: 31.21
USAToday-CNN	% overlapping	s/t: 22.24	t/s: 16.94
	% overlapping (no stop words)	s/t: 15.61	t/s: 12.62
VSoLSCSum	% overlapping	s/c: 37.712	c/s: 44.820

We observed word overlapping between sentences and user posts by considering each token as a single word separated by a space. Statistics from Table 3 show that there exists common words or phrases, which form common hidden topics between sentences and user posts, e.g. 31.21%.

4.2. Document Representation

The first step of a NMF-based summarization system is to convert a document into a matrix, which might be used for decomposition. Given a document d with t terms and n sentences, we can use a term-sentence matrix X to represent d as Gong and Liu, 2001. In this model, $X(i, j)$ is the weight of term t_i in sentence s_j computed by term frequency (TF). However, for our purpose, this representation is deficient since $X(i, j)$ is only calculated on the sentence level, which ignores the document level. We, therefore, adopted an approach of Nakov et al., 2001, which considers both local and global weights of a term.

$$X(i, j) = L(i, j) \times G(i) \tag{1}$$

where $L(i, j)$ is a local weight (TF) and $G(i)$ is a global weight (document inverse frequency - IDF). In other words, Eq. (1) can be referred as the traditional TF-IDF method. This representation is employed for both basic and advanced models given below.

4.3. The Basic Model

We start with a basic model which uses Non-negative Matrix Factorization (NMF)[5] for document summarization. NMF is a useful method based on matrix computation to capture latent structures from non-negative data (Lin, 2007). NMF receives an input matrix X and factorizes X into to matrices W and H so that all elements in three matrices are non-negative. Suppose a data matrix X has a size $n \times m$ with $X_{ij} \geq 0$ and a pre-defined positive integer value $r < \min(n, m)$ (for document summarization, $r = k$ is the number of topics), NMF finds two non-negative matrices $W \in R^{n \times r}$ and $H \in R^{r \times m}$ so that:

$$X \approx WH \tag{2}$$

An usual approach to find W and H is to minimize the difference between

[5] https://en.wikipedia.org/wiki/Non-negative_matrix_factorization

X and WH.

$$\min_{W,H} \qquad f(W,H) \equiv \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}(X_{ij} - (WH)_{ij})^2 \tag{3}$$

$$\text{subject to} \qquad W_{ia} \geq 0, H_{bj} \geq 0 \qquad \forall i, a, b, j. \tag{4}$$

For optimization, Eq. (4) is a standard bound-constrained optimization problem. We also note that the sum part of Eq. (3) can be written as:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}(X_{ij} - (WH)_{ij})^2 = ||X - WH||_F^2 \tag{5}$$

where $|| \cdot ||$ is the Frobenius norm. For optimization, NMF uses an iterative procedure to modify initial values of W and H so that their product approaches A. There are several techniques such as multiplicative update methods (Lee and Seung, 2001), alternating non-negative least squares (Lin, 2007; Lee and Seung, 2001), or gradient approaches (Lin, 2007; Lee and Seung, 2001) which can be employed to optimize Eq. (3). With its nice properties, NMF is usually used for dimensionality reduction, which is useful in many problems such as finding bias vectors of images (Lee and Seung, 1999) or clustering for document summarization (Wang et al., 2008). After factorization, we can use the weights in H to rank sentences and then extract top ranked ones as a summary.

4.4. Advanced Model with Matrix Co-factorization

The basic model uses sentences to create a term-sentence matrix while ignoring the social context of a Web document. We observe that relevant user posts and sentences of a document share common words or phrases, which form common hidden topics. For example, the sentence and comment in Table 1 share common words or phrases, e.g. *"victims"* $\sim$ *"who die"*, *"Germanwings"*, *"families"* $\sim$ *"relatives"*. We define this as *common hidden topics*, which represent the nature of relationships between sentences and user posts. Statistics in Table 3 also support this assumption. We, therefore, introduce a co-factorization method to model common topics between documents and their user posts.

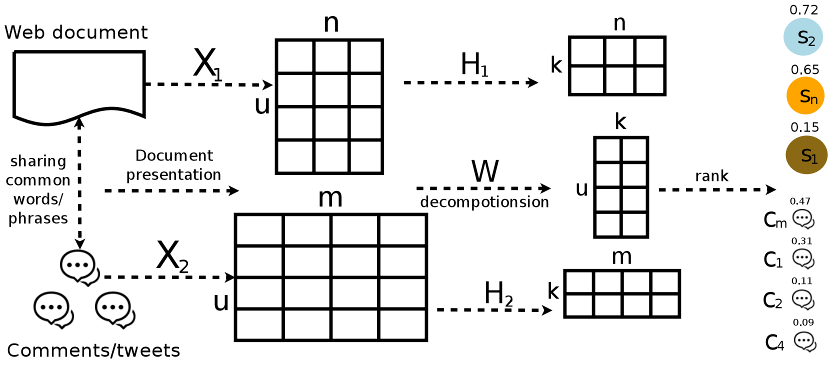


Figure 2: Our co-factorization summarization model.

Figure 2 describes our model. It first maps a document and its user posts into two matrices X_1 and X_2, which share a hidden topic matrix W. Its decomposition next produces two matrices H_1 and H_2, which are analyzed by our new matrix co-factorization algorithm to estimate the importance of sentences and user posts. Our model finally analyzes two column-matrices H_1 and H_2 to extract top m ranked sentences and m ranked user posts as a summary. By introducing a matrix co-factorization model, we can represent and handle the *common topic* hypothesis.

Our model shares the idea of using matrix factorization for summarization with Wang et al., 2008 and Lee et al., 2009; however, we extend this approach to our task by exploiting relevant user posts for single Web document summarization. Our model is different from previous work by using NMF (Wang et al., 2008; Gong and Liu, 2001; Lee et al., 2009; Park et al., 2006) in that it integrates user posts into the ranking process and presents a matrix co-factorization algorithm for extracting summaries. The model is built in two steps: matrix creation and non-negative matrix co-factorization, which are given as follow.

4.4.1. Matrix creation

Given a document d with n main text sentences and m user post sentences, we use two term-sentence matrices: X_1 and X_2 for representing the main text and user posts respectively. To create these matrices, we first merge sentences and user posts into a single set and generate a term dictionary from this set by using word lemmatization, stemming, and stopword removal with `NLTK`,[6].[7] Suppose the size of the dictionary is u, hence we can present $X_1 \in R^{u \times n}$ and $X_2 \in R^{u \times m}$ ($n \ll u$ and $m \ll u$). We apply Eq. (1) to each component of X_1 and X_2 to compute their weights. The TF of a term w is computed on the sentence level, whereas its IDF is calculated on the document level. For example, if w appears in sentences, we consider each sentence as a single document and count the number of sentences including w as its document frequency (DF). If w appears in both document d and its user posts, two DF values are separately calculated for each side. This creation is the first level to take advantage of user posts for creating a term-sentence matrix, which encodes share hidden topics into the scoring step.

4.4.2. Non-negative matrix co-factorization (NMCF)

Our main assumption is that X_1 and X_2 share common hidden topics. Suppose X_1 has k_1 and X_2 has k_2 topics, we distinguish three cases: the same number of topics ($k_1 = k_2 = k$), the number of topics in sentences is larger than that of user posts ($k_1 > k_2$), and reversely, $k_1 < k_2$.[8]

Case 1: $k_1 = k_2 = k$. In this case, we assume that the number of topics in documents and their social context is the same, i.e. it requires that all information posted by readers is relevant to (and cover) the topics in main documents.

Given X_1 and X_2, we can independently apply NMF to X_1 and X_2. However, as they share hidden topics, they can be jointly optimized in an unified co-factorization framework. Let k be the number of the share hidden topics between the main text sentences and user posts of a document d; we can rewrite Eq. (2)

[6] http://www.nltk.org

[7] We did not do lemmatization and stemming on VSoLSCSum since there are no effective algorithms for Vietnamese at the moment.

[8] In the preliminary version of the paper at SoICT'2017, only the first case (($k_1 = k_2 = k$) was considered.

to represent the co-factorization of X_1 and X_2 as follows.

$$X_1 \approx WH_1; \qquad X_2 \approx WH_2 \tag{6}$$

where $W \in R^{u \times k}$ is the matrix of share hidden topics; $H_1 \in R^{k \times n}$ and $H_2 \in R^{k \times m}$ are the sentence and user post weight matrices, which represent the influence of main text and user posts sentences on the hidden topics in W respectively. The sensitivity analysis of selecting topic number k is shown in Section 6.2.

For our co-factorization, the objective function in Eq. (5) can be redefined as follows.

$$E = ||X_1 - WH_1||_F^2 + ||X_2 - WH_2||_F^2 \tag{7}$$

The new objective function includes two components: one for main text sentences and the other one for user posts. The optimization procedure has to consider both components to achieve global (or at least local) optima compared to only optimizing one component as in Eq. (5). The joint optimization in Eq. (7) is the second level of exploiting user posts for our model. Eq. (7) also reveals an important characteristic of our model, that is, sentences and user posts are formulated in a mutual reinforcement support. Eqs. (8)–(9) describe the gradient calculation in our proposed optimization algorithm.

$$\frac{\bigtriangledown E}{\bigtriangledown W} = (WH_1 - X_1)H_1^T + (WH_2 - X_1)H_2^T \tag{8}$$

$$\frac{\bigtriangledown E}{\bigtriangledown H_1} = W^T(WH_1 - X_1); \qquad \frac{\bigtriangledown E}{\bigtriangledown H_2} = W^T(WH_2 - X_2) \tag{9}$$

with update rules as the following:

$$W = W \odot \frac{X_1 H_1^T + X_2 H_2^T}{WH_1H_1^T + WH_2H_2^T} \tag{10}$$

$$H_1 = H_1 \odot \frac{W^T X_1}{W^T W H_1}; \qquad H_2 = H_2 \odot \frac{W^T X_2}{W^T W H_2} \tag{11}$$

where $\odot$ is the Hadamard product. To optimize Eq. (7), we choose to adapt the gradient algorithm (Lin, 2007; Lee and Seung, 2001) for our co-factorization as it has been shown to be efficient in the literature. Our iterative procedure is described in Algorithm 1.

Algorithm 1: Computing error rate in our optimization algorithm.

```
Data: Matrices $W$, $H_1$, and $H_2$.
Result: The weights of these matrices.
1. Initialize $W \geq 0$, $H_1 \geq 0$, and $H_2 \geq 0$ ;
2. for $t = 1, 2, ...$ do
    Update $W, H_1, H_2$ by using Eqs. (10) and (11) ;
    Compute error rate $E$ by using Eq. (7) ;
    if $(E < \epsilon)$ then
        break ;
    end
end
Output: $W, H_1$, and $H_2$ ;
```

Our algorithm first initializes W, H_1, and H_2 with a constraint that all values ≥ 0. In each iteration, it computes the entries of W, H_1, and H_2 based on Eqs. (10) and (11) and then calculates an error value in Eq. (7). The algorithm stops if the error value $\leq \epsilon$ or the number of iterations exceeds a certain value. In our experiments, we fix $\epsilon = 0.01$ and $inter = 1000$. In practice, to ensure the uniqueness of our NMCF, we normalize W, H_1, and H_2 with L_1 or L_2.

Normalization of L_1:

$$w_{ij} = \frac{w_{ij}}{\sum_i w_{ij}}; \qquad h_{ij} = h_{ij} \sum_i w_{ij} \tag{12}$$

or L_2:

$$w_{ij} = \frac{w_{ij}}{\sqrt{\sum_i w_{ij}^2}}; \qquad h_{ij} = h_{ij} \sqrt{\sum_i w_{ij}^2} \tag{13}$$

The effects of using L_1 or L_2 are analyzed in §6.4.

***Case 2:* $\boldsymbol{k}_1 > \boldsymbol{k}_2$.** The model in Eq. (7) strictly requires X_1 and X_2 sharing a topic matrix W, which has the same topic number. It is somewhat uncommon because, in practice, the topic number of documents and their user posts may be different. We, therefore, relax our NMCF in Eq. (7) by considering the number of topics in documents is larger than that of their user posts.

Suppose the share matrix $W \in R^{u \times k_1}$, $H_1 \in R^{k_1 \times n}$, and $H_2 \in R^{k_2 \times m}$, the representation of X_1 and X_2 in Eq. (6) can be rewritten as follows.

$$X_1 \approx WH_1; \qquad X_2 \approx WIH_2 \tag{14}$$

where $I \in R^{k_1 \times k_2}$ is a matrix with values on the diagonal line $= 1$. The intuition behind this formulation is that the number of topics of user posts is a subset of topics in sentences. Therefore, we can denote $k_2 \subset k_1$ and the topic matrix of X_1 is $W \in R^{n \times k_1}$. As our assumption, we can pick first k_2 vectors of W to create the topic matrix of X_2. This operation creates the topic matrix of X_2 as WI, with $I \in R^{k_1 \times k_2}$. As a result, the matrix of user posts can be written as $X_2 \approx WIH_2$. Values on the diagonal line of the matrix I equals 1 because we want to keep the same weights of W for both X_1 and X_2. The new objective function is now re-defined as Eq. (15).

$$E = ||X_1 - WH_1||_F^2 + ||X_2 - WIH_2||_F^2 \tag{15}$$

followed by the gradient calculation of optimization.

$$\frac{\bigtriangledown E}{\bigtriangledown W} = (WH_1 - X_1)H_1^T + (WIH_2 - X_2)(IH_2)^T \tag{16}$$

$$\frac{\bigtriangledown E}{\bigtriangledown H_1} = W^T(WH_1 - X_1); \qquad \frac{\bigtriangledown E}{\bigtriangledown H_2} = W^T(WIH_2 - X_2) \tag{17}$$

with update rules as the following:

$$W = W \odot \frac{X_1 H_1^T + X_2(IH_2)^T}{WH_1H_1^T + WH_2(IH_2)^T} \tag{18}$$

$$H_1 = H_1 \odot \frac{W^T X_1}{W^T W H_1}; \qquad H_2 = H_2 \odot \frac{(WI)^T X_2}{(WI)^T (WI) H_2} \tag{19}$$

We can observe that X_2 now has an additional element I, which denotes the relationship between k_1 and k_2 while X_1 is the same with the original NMCF model in Eq. 7. Update rules also consider I as an aspect of the optimization algorithm. This model also uses normalization by using L_1 and L_2. We also apply Algorithm 1 to optimize this model.

Case 3: $\boldsymbol{k}_1 < \boldsymbol{k}_2$. In this case, we consider the topic number in a document is smaller than in user posts. The intuition of our consideration is that topics in sentences are a subset of topics in user posts. Similar to case 2, we introduce a new matrix I in the place of matrix X_1, the representation can be rewritten as follows:

$$X_1 \approx WIH_1; \qquad X_2 \approx WH_2 \tag{20}$$

Consequently, the new objective function is defined as:

$$E = ||X_1 - WIH_1||_F^2 + ||X_2 - WH_2||_F^2 \tag{21}$$

By following definitions in Eqs. (16)–(17), we can move the matrix I from X_2 to X_1 to define new gradient based optimization algorithm with the new update rules in the same mechanism. The objective function in Eq. (21) is also optimized by using Algorithm 1. After optimizing, weights in H_1 and H_2 are used to compute the rank of sentences and user posts shown in the next section.

4.5. Sentence Selection

After finding (local) optimal solutions, we compute the weights in H_1 for ranking sentences and calculate the weights in H_2 for ranking user posts. The weight, WS_j, of each sentence (or each user post) is calculated as follows.

$$WS_j = \sum_{i=1}^{k} H_{ij} \times weight(H_i) \tag{22}$$

$$weight(H_i) = \frac{\sum_{q=1}^{n} H_{iq}}{\sum_{p=1}^{k} \sum_{q=1}^{n} H_{pq}} \tag{23}$$

where k is a chosen topic number and n is the number of sentences (or user posts). When all sentences are weighted with WS_j, we can rank them based on their scores on the matrix H_1 and rank user posts based on their weights on the matrix H_2. We employ a simple greedy algorithm to extract summaries. After scoring and ranking, our model loops on ranked sentences (and user posts), dequeues one sentence (and user post), and puts it into a summary. This process stops when the number of selected sentences (or user posts) reaches to m. We also apply Eqs. (22) and 23 to the basic NMF model.

5. Statistical Analysis

5.1. Settings

We used 10-fold cross-validation for SoLSCSum with $m = 6$ as the setting of Nguyen et al., 2016c. For USAToday-CNN and VSoLSCSum, we used 5-fold cross-validation as suggested in Nguyen et al., 2016a and Wei and Gao, 2014. We fix $m = 4$ for USAToday-CNN because each document has 3-4 highlights and $m = 6$ for VSoLSCSum as the same setting of Nguyen et al., 2016a. Stop words and links were removed, except for VSoLSCSum due to there is no formal stop word list for Vietnamese.

5.2. Baselines

We validate the efficiency of our model by comparing with baselines of text and social context summarization. They are in three groups: NMF methods, non-social context methods, and social context methods.

NMF methods. We compare our model to the basic method using **NMF**. We expect that with the support of user posts in a joint optimization algorithm, our model significantly outperforms the basic one.

Non-social context methods. These methods do not use the support of social formation. **Lead-m** chooses the first m sentences as a summary (Nenkova, 2005). This method is not used to select tweets or comments. **LexRank**[9] (Erkan and Radev, 2004) builds a stochastic graph for computing relative importance of textual units in text summarization. **SVM** (Cortes and Vapnik, 1995) is used by Yang et al., 2011 to train a binary classifier, which is applied to testing data to extract summaries. To implement this method, we use LibSVM[10] with the RBF kernel and five basic features in Yang et al., 2011.

Social context methods. They exploit the support of social information in the summarization process. **cc-TAM** uses a cross-collection topic-aspect model (cc-TAM) as a preliminary step to generate a bipartite graph used for co-ranking (Gao et al., 2012). **RankBoost CCF** uses many sophisticated local and cross features to train a learning to rank model for ranking sentences and tweets (Wei and Gao, 2014). **HGRW** is a variation of LexRank named Heterogeneous Graph Random Walk (Wei and Gao, 2015). The authors integrate tweets into the calculation of LexRank. **SoRTESum** uses one wing (sentences) or dual wing information (sentences and tweets) (Nguyen and Nguyen, 2016). It contains two methods: using inter information (SoRTESum Inter Wing) and dual wing information (SoRTESum Dual Wing). **SVM Ranking** is recently used in Nguyen et al., 2016b for summarization. This method is an extension of RankBoost CCF, in which the authors include new features and employ a different ranking method (Ranking SVM).

5.3. Evaluation Metrics

For USAToday-CNN, we used highlights as gold-standard references. For SoLSCSum and VSoLSCSum, selected sentences and comments (labeled by 1 in the annotation step) were used as gold-standard references. We used ROUGE-

[9] https://code.google.com/p/louie-nlp/source/browse/trunk/louie-ml/src/main/java/org/louie/ml/lexrank/?r=10

[10] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

scores (Lin and Hovy, 2003) for our evaluation.

$$ROUGE-N = \frac{\sum_{s \in S_{ref}} \sum_{gram_n \in s} Count_{match}(gram_n)}{\sum_{s \in S_{ref}} \sum_{gram_n \in s} Count(gram_n)} \tag{24}$$

where n is the length of n-gram, $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and the references, $Count(gram_n)$ is the number of n-grams in the references. In practice, we employed `ROUGE-1.5.5` by using `pyrouge`[11] with parameters `‘‘-c 95 -2 -1 -U -r 1000 -n 2 -w 1.2 -a -s -f B -m"`. We used the F-score of ROUGE-1, ROUGE-2, and ROUGE-W as major metrics to balance the different length of summaries and references.

6. Results and Discussion

This section first shows ROUGE-scores of our model against the baselines and refined methods. It next presents observation for investigating several aspects of our model with discussions.

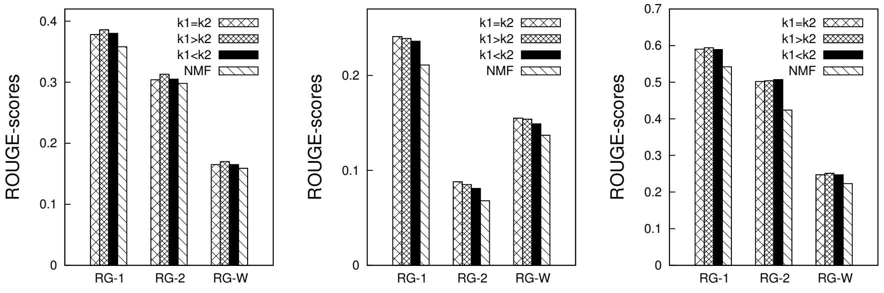
6.1. ROUGE-scores

This section shows our comparison in three parts: NMCF vs. NMF, comparison with non-social context and with social context methods.

***NMCF** vs. **NMF**.* We first compared our NMCF to the basic model based on NMF. Figures 3 and 4 summarize ROUGE-scores of our comparison. It is clear from these figures that our NMCF obtains sufficient improvements over NMF in almost all cases (p-values ≤ 0.05). For example, for sentence extraction on VSoLSCSum in Figure 3c, the performance of our model is 5% higher of ROUGE-1 and 7% of ROUGE-2 than NMF. This trend is consistent in extracting comments in Figure 4, in which there are big margins between our methods and NMF. This confirms the efficiency of our model in exploiting mutual information between user posts and sentences. In Figure 3a, NMCF slightly outperforms NMF, e.g. 0.378 vs. 0.358 of ROUGE-1. This is because NMF

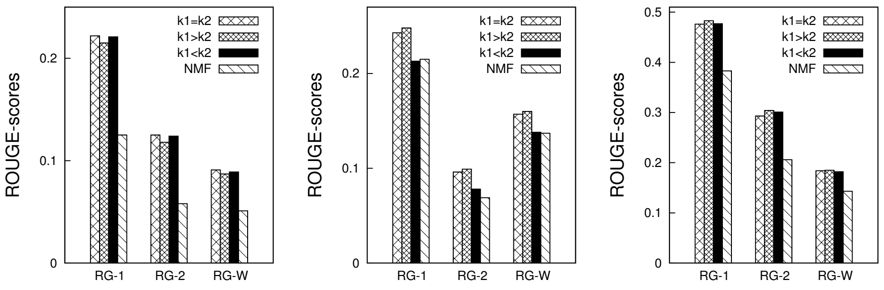
[11] https://github.com/andersjo/pyrouge

also utilizes the advantage of matrix factorization, which has shown to be efficient for document summarization (Gong and Liu, 2001; Wang et al., 2008; Lee et al., 2009; Park et al., 2006). Also, sentences themselves contain important information; hence, adding more data from user posts slightly improves ROUGE-scores. However, for comment extraction, the support from sentences boosts ROUGE-scores of our model with large margins compared to NMF.



(a) SoLSCSum (b) USAToday-CNN (c) VSoLSCSum

Figure 3: NMCF vs. NMF for sentence selection.



(a) SoLSCSum (b) USAToday-CNN (c) VLSCSum

Figure 4: NMCF vs. NMF for user post extraction.

We conduct statistical tests (the pairwise $t-$test) to confirm the significance of these improvements. p-values in Table 4 statistically confirm the efficiency of our model, in which it significantly outperforms NMF in almost all cases. For sentence extraction on SoLSCSum, the ROUGE-1 of NMF is competitive with our model with $p-$value = 0.2868.

Table 4: The pairwise t-test of NMCF and NMF. **Bold** is significant with $p \leq 0.05$.

Data	Method	Sentence			User posts		
		RG-1	RG-2	RG-W	RG-1	RG-2	RG-W
SoLSCSum	$k_1 = k_2$	0.2868	0.3277	0.6711	**0.0000**	**0.0000**	**0.0000**
	$k_1 > k_2$	0.1374	0.1018	0.4011	**0.0000**	**0.0000**	**0.0000**
	$k_1 < k_2$	0.2806	0.3362	0.6615	**0.0000**	**0.0000**	**0.0000**
USA-CNN	$k_1 = k_2$	0.1361	0.1609	0.2013	**0.0143**	**0.0140**	**0.0037**
	$k_1 > k_2$	0.1607	0.2499	0.1939	**0.0057**	**0.0092**	**0.0017**
	$k_1 < k_2$	0.2115	0.3857	0.3678	—	0.2617	0.8512
VSoLSCSum	$k_1 = k_2$	**0.0384**	**0.0008**	**0.0000**	**0.0006**	**0.0000**	**0.0296**
	$k_1 > k_2$	**0.0507**	**0.0017**	**0.0000**	**0.0002**	**0.0000**	**0.0223**
	$k_1 < k_2$	0.0631	**0.0004**	**0.0001**	**0.0003**	**0.0000**	**0.0333**

Comparison with non-social context methods. We report the comparison of our model with non-social context methods in Table 5. ROUGE-scores indicate that our NMCF outperforms strong methods in almost all cases.[12] For example, it obtains very competitive ROUGE-scores on SoLSCSum and VSoLSCSum, and it is the best in extracting user posts on USAToday-CNN. There are big margins between our methods and the second best method, e.g. 0.590 vs. 0.506 (significant improvements are denoted by † with $p-$values ≤ 0.05). This trend is consistent in other cases. This is because our methods utilize the support of social context and take advantage of matrix co-factorization. For the first aspect, user posts help to enhance the information of sentences in document representation. A bigger dictionary created from sentences and user posts enriches the representation of term-sentence matrices, which improves the quality of the scoring step. For the second aspect, scoring with matrix co-factorization exploits the share topic matrice to effectively rank sentences and user posts. This draws a similar conclusion with Yang et al., 2011; Wei and Gao, 2014;

[12] There are slight differences between ROUGE-scores in this manuscript and the original paper because we here use L_1 rather than L_2.

Table 5: NMCF vs. basic methods; **bold** is the best value; *italic* is the second best. RG stands for ROUGE. † shows that our methods obtain significant improvements.

Dataset	Method	Sentences			User posts		
		RG-1	RG-2	RG-W	RG-1	RG-2	RG-W
SoLSCSum	Lead-m	0.345	**0.322**	**0.170**	—	—	—
	LexRank	0.327†	0.243†	0.138†	0.210	0.115	0.085
	SVM*	0.325†	0.263†	0.147	0.152†	0.089†	0.062†
	NMCF ($k_1 = k_2$)	0.378	0.304	*0.165*	**0.222**	**0.125**	**0.091**
	NMCF ($k_1 > k_2$)	**0.386**	*0.313*	**0.170**	0.215	0.118	0.087
	NMCF ($k_1 < k_2$)	*0.380*	0.305	*0.165*	*0.221*	*0.124*	*0.089*
USAToday-CNN	Lead-m	0.249	**0.106**	**0.172**	—	—	—
	LexRank	*0.251*	*0.092*	0.163	0.193†	0.068†	0.128†
	SVM*	**0.261**	**0.106**	*0.171*	0.221	0.084	0.149
	NMCF ($k_1 = k_2$)	0.241	0.088	0.155	*0.243*	*0.096*	*0.157*
	NMCF ($k_1 > k_2$)	0.239	0.085	0.154	**0.248**	**0.099**	**0.160**
	NMCF ($k_1 < k2$)	0.236	0.081	0.149	0.213	0.078	0.138
VLSCSum	Lead-m	0.495†	0.420†	0.214†	—	—	—
	LexRank	0.506†	0.432†	0.219†	0.348†	0.198†	0.127†
	SVM*	0.497†	0.440†	0.208†	0.374†	0.212†	0.140†
	NMCF ($k_1 = k_2$)	*0.590*	0.502	*0.247*	0.476	0.293	*0.184*
	NMCF ($k_1 > k_2$)	**0.594**	*0.504*	**0.251**	**0.483**	**0.304**	**0.185**
	NMCF ($k_1 < k_2$)	0.589	**0.507**	*0.247*	*0.477*	*0.301*	0.182

Li et al., 2016, in which social information can support sentences to improve the quality of scoring if such information is exploited in an appropriate way. For sentence selection, our method is competitive on USAToday-CNN. SVM and Lead-m achieve the best results because SVM is also a supervised method, which uses several suitable features. Lead-m outperforms many systems participated in DUC (Nenkova, 2005) because it simulates the writing style of humans by putting important information in first sentences. However, in other cases, our NMCF surpasses these methods.

Table 6: ROUGE-scores of our model and advanced methods.

Dataset	**Method**	**Sentences**			**User posts**		
		RG-1	RG-2	RG-W	RG-1	RG-2	RG-W
SoLSCSum	cc-TAM	0.306†	0.238†	0.136†	0.054†	0.022†	0.024†
	RB* CCF	0.360	0.283	0.158	0.190†	0.098	0.077†
	HGRW	0.379	0.204	0.167	0.209	0.115	0.084
	Inter-Wing	0.352	0.277	0.154	0.209	0.115	0.083
	Dual-Wing	0.357	0.280	0.156	0.180	0.098	0.070
	SVMRank*	**0.386**	0.291	**0.178**	0.207	0.103	0.088
	NMCF ($k_1 = k_2$)	0.378	0.304	0.165	**0.222**	**0.125**	**0.091**
	NMCF ($k_1 > k_2$)	**0.386**	**0.313**	*0.170*	0.215	0.118	0.087
	NMCF ($K_1 < k_2$)	*0.380*	*0.305*	0.165	*0.221*	*0.124*	*0.089*
USAToday-CNN	cc-TAM	0.229	0.077	0.145	**0.249**	0.089	0.152
	RB* (CCF)	0.221	0.070	0.140	0.233	0.091	0.132
	HGRW	**0.279**	*0.098*	**0.177**	0.242	0.088	*0.157*
	Inter-Wing	*0.268*	**0.104**	*0.173*	0.215	0.074	0.137
	Dual-Wing	0.252	0.085	0.163	0.227	0.080	0.146
	SVMRank*	0.240	0.084	0.153	0.217	0.080	0.140
	NMCF ($k_1 = k_2$)	0.241	0.088	0.155	0.243	*0.096*	*0.157*
	NMCF ($k_1 > k2$)	0.239	0.085	0.154	*0.248*	**0.099**	**0.160**
	NMCF ($k_1 < k2$)	0.236	0.081	0.149	0.213	0.078	0.138
VLSCSum	cc-TAM	0.488†	0.377†	0.201†	0.301†	0.167†	0.111†
	RB* CCF	0.561	0.494	0.235†	0.471	*0.308*	0.168
	HGRW	0.570	0.479	0.233†	0.454	0.298	0.173
	Inter-Wing	0.532†	0.463†	0.218†	0.443	0.277†	0.170
	Dual-Wing	0.531†	0.457†	0.218†	0.409†	0.234†	0.153
	SVMRank*	0.582	**0.527**	*0.249*	*0.482*	**0.319**	0.183
	NMCF ($k_1 = k_2$)	*0.590*	0.502	0.247	0.476	0.293	*0.184*
	NMCF ($k_1 > k_2$)	**0.594**	*0.504*	**0.251**	**0.483**	*0.304*	**0.185**
	NMCF ($k_1 < k_2$)	0.589	*0.507*	0.247	0.477	0.301	0.182

Comparison with social context methods. We report the comparison of our model and social context methods in Table 6. Again, ROUGE-scores from this table validate the efficiency of our model, in which it is the best in many cases. Our methods obtain competitive results on SoLSCSum, in which it is the best except for ROUGE-W of sentence selection. SVMRank and HGRW also perform comparably with our model. It is understandable that SVMRank uses many domain-dependent features to model sentence-comment relationships in the form of supervised learning, which leads to a strong method. However, our model is unsupervised, which is domain-independence. For HGRW, it exploits user posts to score sentences by using a random walk ranking algorithm (Wei and Gao, 2015). However, our methods are still better than these methods in many cases. ROUGE-scores of cc-TAM are quite poor because it is designed for multi-document summarization (Gao et al., 2012) while all the datasets are for single-document summarization. The trend is consistent on VSoLSCSum, in which our methods are the best, followed by SVMRank.

Our methods achieve comparable results for sentence selection on USAToday-CNN. HGRW and SoRTESum Inter-Wing obtain the best scores. In this sense, our model is limited to the abstract aspect of highlights, which can be addressed by features of SoRTESum Inter-Wing. This also again confirms that HGRW is a strong method even it is unsupervised learning. Note that SVMRank does not achieve the best results even it is a supervised learning method. On the other hand, our model obtains promising results of tweet extraction, in which it is competitive in several cases. Interestingly, cc-TAM is the best of ROUGE-1 for tweet extraction, but our model can still compare to cc-TAM with a tiny margin, i.e. 0.248 vs. 0.249.

ROUGE-scores on DUC 2004. We confirmed NMCF's efficiency on DUC 2004. Our objective is to show the adaptation of our model on a standard dataset of text summarization rather than obtaining the best results on this dataset, which lacks social information. Also, we would like to observe margins between our NMCF and advanced methods. Since there are tiny margins among our methods; we, hence, only show results of NMCF with the same topic number.

DUC 2004 contains 50 topics, in which each topic has 10 articles and four references written by humans. Since it has no social information, we adapted it to our model by using a one-versus-all setting. We kept one article as a primary document and formed nine remaining ones as relevant information. The intuition of this setting is that our model can be extended to exploit relevant articles of main documents for single document summarization. We applied NMCF to each primary document to select two sentences, resulting 20 sentences in total. To select summaries, we employed a simple greedy algorithm. Sentences fewer than five words were first removed because they are fairly short for summarization (Erkan and Radev, 2004) and the rest were sorted in decreasing order based on their Cosine scores.

$$score(s_i) = \frac{1}{|s_i|} \sum_{j=1}^{m} cos(s_i, s_j) \tag{25}$$

where m is all other sentences in a topic. We iteratively dequeue one sentence from the sorted list and append it to form a summary if it is non-redundant with a Cosine threshold = 0.75. The iteration stops if the number of selected sentences reaches a length constraint.

We report three basic methods: PROB, LLR, MRW presented in (Hong and Nenkova 2014) and three advanced methods: (i) MD-ILP, an abstractive ILP-based summarizer (Banerjee et al., 2015); (ii) REGSUM, a regression-based model with hand-crated features (Hong and Nenkova 2014); and (iii) CRSum, a deep learning model based on the sentence context (Ren et al., 2017) with ROUGE-1, 2, and 4 recall. Due to the setting of DUC, the evaluation uses a length constraint with a parameter `‘‘b 665’’` (665 bytes).

ROUGE-scores in Table 7 indicate that our method outputs promising results on DUC 2004. It outperforms three basic models: PROB, LLR, MRW. The ROUGE-scores of NMF are also competitive, but our method is still better than NMF. However, there are rather gaps between NMCF and state-of-the-art methods. For example, CRSum is the best of ROUGE-1 and REGSUM is the best of ROUGE-4. This is understandable that they are supervised learn-

Table 7: ROUGE-scores on DUC 2004.

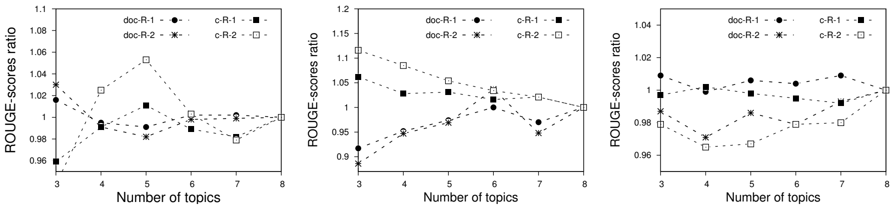
Method	ROUGE-1	ROUGE-2	ROUGE-4
PROB†	0.3514	0.0817	0.0106
LLR†	0.3460	0.0756	0.0083
MRW†	0.3578	0.0815	0.0099
MD-ILP†	—	**0.1199**	—
REGSUM†	*0.3857*	0.0975	**0.0160**
CRSum†	**0.3953**	*0.1060*	—
NMF	0.3557	0.0762	0.0107
NMCF	0.3734	0.0846	*0.0132*

ing methods. CRSum exploits the context surrounding a sentence in word and sentence levels, respectively. The final vector representation of a sentence is concatenated with several surface features such as its position. REGSUM uses sophisticated hand-crafted features to estimate the importance of words, which can be used to measure the importance of sentences. By contrast, our model is unsupervised learning developed for social context summarization but not for multiple-document summarization. In another case, MD-ILP is the best of ROUGE-2 because it exploits informativeness and linguistic aspects with a set of constraints. However, adapting these methods to our task is still an open question. On the other hand, our method can be flexibly adapted to domains which include user posts (Figures 3, 4, and Tables 5, 6) or those which do not include user posts as DUC 2004 in Table 7 with very competitive results.

6.2. Topic Analysis

In this subsection, we analyze the influence of topic numbers on our model with three aforementioned settings.

Case 1: the same number of topics. We consider the sensitivity analysis of k in our model by tuning the topic number k in [3, 8] with a jumping step = 1. The number of topics outside this rage is too small or large. More precisely, we normalized Figure 5 by dividing ROUGE-scores of each tuning point for those at $k = 8$ to handle small margins among tuning points.



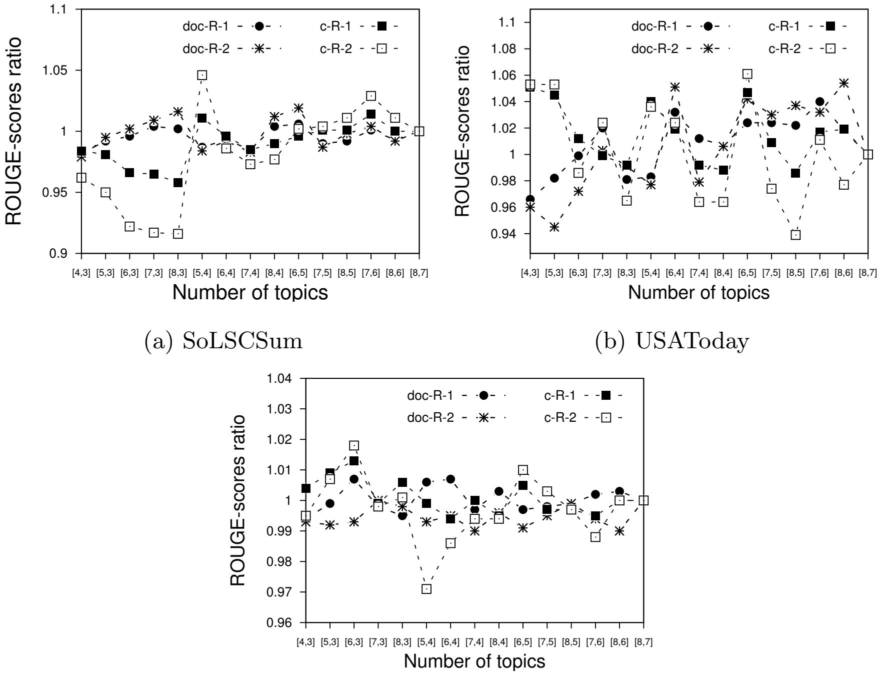
(a) SoLSCSum (b) USAToday (c) VSoLSCSum

Figure 5: ROUGE-scores with various k.

Figure 5 shows that the topic number k affects our model. In Figure 5a, increasing k reduces ROUGE-scores. Our model reaches a peak at $k = 5$; then its performance slightly decreases at $k = 8$. This trend is quite similar to Figure 5b in which our model obtain better results with smaller topic numbers. By contrast, in Figure 5c, the trend is reverse, in which increasing k improves ROUGE-scores. The trend in Figure 5b is inconsistent, in that increasing k improves ROUGE-scores of sentence selection, but the performance of tweet extraction decreases. The general trend from these figures indicates that our model operates well at $k = 5, 6$. Gaps between lower and upper bounds are small (around 0.05) showing that changing k slightly influences ROUGE-scores.

Case 2: $\boldsymbol{k}_1 > \boldsymbol{k}_2$. This setting considers the number of topics in a document is larger than that in its user posts. To investigate the influence of k_1 and k_2, we tuned k_1 in [4, 8] and k_2 in [3, 7] so that $k_1 > k2$. In each pair of tuning points, we observed its ROUGE-scores and plotted them on Figure 6.

From Figure 6 we see that all lines are fluctuant. The trend in Figure 6a seems to increase while it tends to decrease in Figure 6c. In Figure 6b, it is hard to conclude the general direction. However, our model obtains the best results with $k_1 = 6$ and $k_2 = 5$. For some cases in which the number of topics is very different, e.g. [8, 3], results are not as good as pairs which have similar topics such as [6, 5]. A possible reason is that our model divides documents into too



(a) SoLSCSum

(b) USAToday

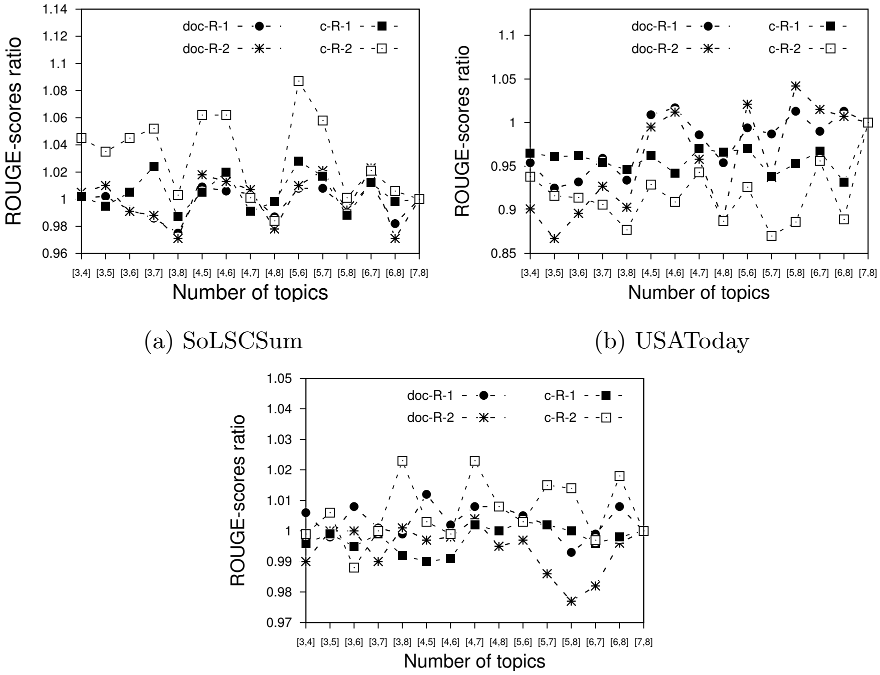
(c) VSoLSCSum

Figure 6: ROUGE-scores with various k pairs with $k_1 > k_2$.

specific topics while it splits user posts into too general ones, which bias the weight computation of H_1 and H_2. This also explains that the model with the same topic number ($k = 5, 6$) can achieve competitive scores in several cases.

Case 3: $\boldsymbol{k_1 < k_2}$. We also observed the influence of topic number in the setting of $k_1 < k_2$. We used the same tuning procedure in the second case but changed the constraint which requires $k_1 < k_2$. We visualize ROUGE-scores after normalizing on Figure 7.

The observation is similar to the second case in that our model outputs better results with topic pairs which are close in term of numbers. Therefore, to balance ROUGE metrics, we suggest that pairs created by combining values in [4, 5, 6] can be used.



(a) SoLSCSum

(b) USAToday

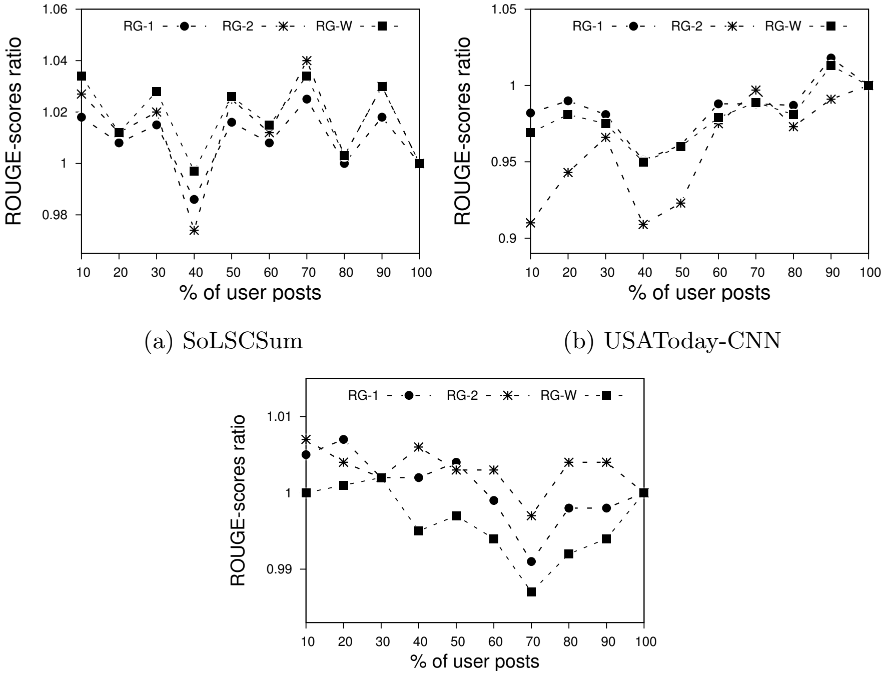
(c) VSoLSCSum

Figure 7: ROUGE-scores with various k pairs.

6.3. The Number of User Posts

We investigated the contribution of user posts based on an assumption that the number of user posts affects our model. To do that, we adjusted the number of user posts of each main document from 10% to 100% based on their order. The intuition behind this is that we would like to simulate the nature of user posts, which come in an accumulative fashion. After dividing, we run our model with this segmentation and observe its ROUGE-scores. We normalized the ROUGE-scores by dividing the scores of each data-size point to those of 100% to handle tiny margins among tuning points. It is possible to show all methods but here we report our model with k=5 using L_1 due to space limitation.

Figure 8 shows the trend of ROUGE-scores of sentence selection. This indicates that the number of user posts influences our model. The trend in Figure 8a fluctuates in which it decreases from 10% to 40% and reaches a peak at 70%.



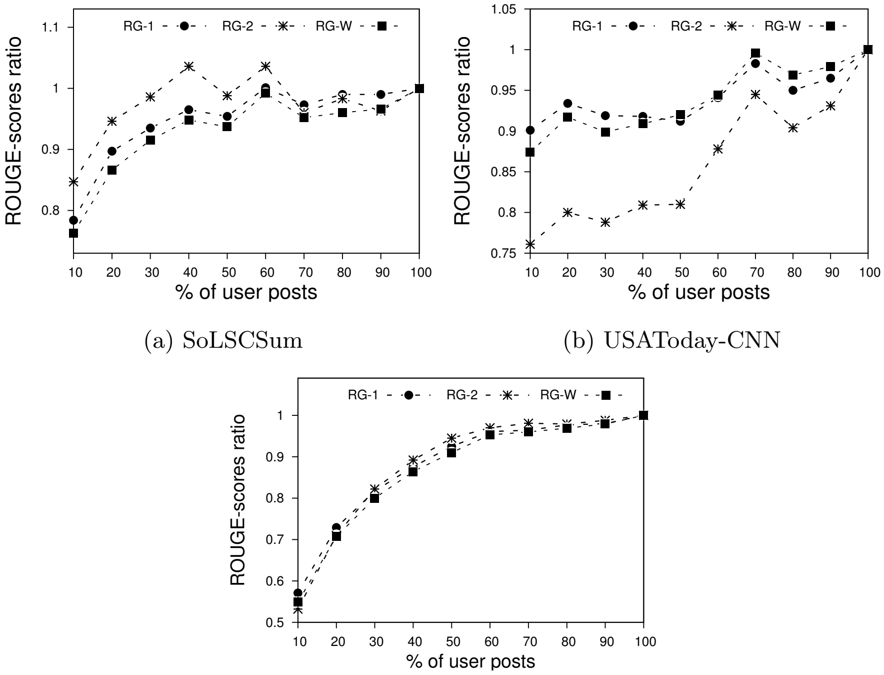
(a) SoLSCSum

(b) USAToday-CNN

(c) VSoLSCSum

Figure 8: The contribution of user posts for sentence selection.

After that it slightly falls. The trend in Figures 8b and 8c is reverse. By increasing the number of user posts, the performance of our model also increases on USAToday-CNN. For VSoLSCSum, ROUGE-scores tend to decrease until the point of 70%. After that, it raises again. The general trend in Figure 8 reveals two interesting points. Firstly, our model does not obtain the best results by using all user posts (100%). This is because of the nature of user posts, which are posted in an accumulative fashion, after the appearance of main events. In this way, user posts in some time steps include salient information, which contributes our model, e.g. 70% in Figure 8a. After that, users can post noise messages, which challenge our model, e.g. 80% in Figure 8a or 70% in Figure 8c. Secondly, small margins between maximal and minimal values in these figures indicate that user posts contribute our model; however, they are less important than sentences. It is understandable that sentences include important information and our model exploits user posts as an additional data channel. This also supports results in Figure 3, in which our model slightly outperforms NMF, which only uses sentences.



(a) SoLSCSum

(b) USAToday-CNN

(c) VSoLSCSum

Figure 9: The contribution of user posts for user post extraction.

For user post extraction, the general trend in Figure 9 is reverse compared to the tendency in Figure 8. By increasing the number of user posts, the performance of our model also increases. This is because the small number of user posts challenges our model to extract informative user posts, even our model exploits sentences to enrich the representation of matrices. To balance the performance of sentence selection and user post extraction, we use all user posts for supporting sentences in our model.

6.4. Normalization Observation

As mentioned, our model uses L_1 or L_2 as normalization to avoid over-fitting during optimization. Due to tiny margins among our methods, we here report ROUGE-scores of NCMF with the same topic number in Table 8.

Table 8: Norm observation. RG means ROUGE-scores.

Dataset	**Method**	**Sentences**			**User posts**		
		RG-1	RG-2	RG-W	RG-1	RG-2	RG-W
SoLSCSum	L_1	0.378	0.304	0.165	0.222	0.125	0.091
	L_2	0.387	0.312	0.168	0.218	0.117	0.088
USAToday-CNN	L_1	0.241	0.088	0.155	0.243	0.096	0.157
	L_2	0.212	0.069	0.136	0.237	0.092	0.155
VLSCSum	L_1	0.590	0.502	0.247	0.476	0.293	0.184
	L_2	0.590	0.505	0.247	0.483	0.306	0.186

From this table we can observe that normalization with L_2 outputs better results than L_1 for sentence selection on SoLSCSum and on VSoLSCSum. For example, ROUGE-scores of L_2 are higher than those of L_1 on VSoLSCSum. For comment extraction on SoLSCSum, L_1 is better. On USAToday-CNN, L_1 consistently surpasses L_2. To balance performance on three datasets, we selected L_1 as normalization for our methods.

6.5. Error Rate in Optimization

We observed the trend of error reduction in our optimization algorithm. Figure 10 shows that the error is asymptotic to 0 when the number of iterations get increased. It is significantly reduced after 200 iterations. After that, it slightly falls and is nearly 0 after 1000 iterations. It is understandable that the optimization algorithm finds a better optimal solution with a large number of iterations, but it takes a long time for convergence. Based on this observation, we fix the iterations of the algorithm within 1000.

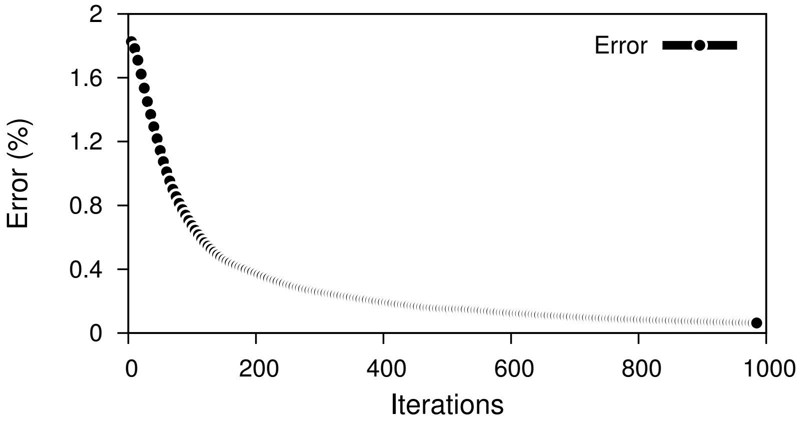


Figure 10: The convergence of the optimization algorithm.

6.6. Output Analysis

We examined extracted sentences and comments of our model and NMF on SoLSCSum. From Table 9, we can observe that our model extracts one correct sentence and comment. The sentence locates in the second position. It contains much important information of the event *"Germanwings crash families could seek damages in the U.S.: lawyer"* such as the type of event (*"Germanwings crash"*), the situation (deal with the company for the damage). In many cases, reading S1 can provide enough salient information about this event. The comment C1 also reflects the content of this event. By reading C1, we can partly understand the event, in which the relatives of victims try to claim with the company for their money. Interestingly, C1 includes reader's opinions regarding victims. However, topics mentioned in comments are usually diverse, then they bring the noise, such as C2. Our model also extracts incorrect sentences (S2 and C2). While S2 is relevant to the event, it is hard to conclude C2 belongs to this event. However, they include many words, which appear in summary sentences, such as *"airline"*, *"US"*, or *"claim"*. Also, as our expectation, extracted sentences share many common words because we present the nature of sentences and user posts in a share hidden topic-matrix. In this case, many common words help to improve the representation of a document.

Table 9: A summary example generated from the document 31^{st} on SoLSCSum; two summary sentences and comments are shown instead of six. Sentences with [+] means that they are also in the references; [-] means that they are not in the references.

NMCF Summarization (the same topic number)
Sentence selection
[+]S1: Families of the victims of the Germanwings crash are considering filing a claim for damages in the United States if they cannot reach agreement with parent airline Lufthansa in Germany, a lawyer representing the families said on Sunday.
[-]S2: “If the airline is not prepared to do so, however, we will look seriously at making a claim in the United States,” said Giemulla, adding that he was representing 21 families including those of the German school children who died.
Comment extraction
[+]C1: I’m sad for the people who are no longer on their life of this Germanwings strategy, but I don’t know how much money relatives want for the flesh of who die on Germanwings case.
[-]C2: The only ones who should be allowed to take this to a US court are the US citizens.
NMF Summarization
Sentence selection
[+]S1: Families of the victims of the Germanwings crash are considering filing a claim for damages in the United States if they cannot reach agreement with parent airline Lufthansa in Germany, a lawyer representing the families said on Sunday.
[-]S2: Nearly half of the victims of the Germanwings Barcelona to Duesseldorf flight were German, with the remaining passengers hailing from a range of countries, including Spain, Australia and Argentina.
Comment extraction
[-]C1: Julie to Ann (who is holding a cabbage): “I would have taken a ride with anyone except Peter Kramer, Freddiemaybe Joe.
[-]C2: Don’t understand why a US court would have authority over European citizens.

NMF shares one correct sentence (S1) with our model. It is understandable that NMF is also competitive in summarizing documents. It selects the second one, which is relevant to the event, but not important at this moment because the event passed. This explains that our model outperforms NMF in sentence selection in Section 6.1. For comment extraction, NMF extracts incorrect comments (two of those are shown in Table 9). This explains the reason that our model significantly surpasses NMF in Figure 4a. Extracted sentences and comments share few common words because it does not exploits the share of common topics in the summarization process.

7. Conclusion

This paper presents a model, which exploits user posts such as comments and tweets to enrich Web document summarization. The insight behind our model comes from the fact that sentences and user posts share hidden topics denoted in the form of common words or phrases. Our model captures mutual information between sentences and user posts by assuming they share hidden topics which can be found by a matrix co-factorization approach. The decomposition extracts salient sentences and user posts, which have two characteristics: (i) reflecting document content and (ii) sharing common topics. Applying our model to the task of sentence and highlight extraction of single documents indicates that it can be viable alternative to extraction-based systems. ROUGE-scores on three datasets in two languages, English and Vietnamese, of social context summarization and DUC 2004 confirm the efficiency of our NMCF. The model achieves promising results in an unsupervised fashion, without reference to any NLP tools (e.g. parsing) suggesting that it can be applied to unrestricted domains.

For future directions, an obvious next step is to investigate how the model works to other domains and text genres which include user posts. The document representation in Section 4.2 can also be presented by using semantic levels such as word or sentence embedding.

Acknowledgment

This work was supported by JSPS KAKENHI Grant numbers 15K16048 and JP15K12094; and UTEHY.T026.P1718.04. We also thank comments of reviewers and anonymous readers for improving our paper.

References

Amitay, E., & Paris, C. (2000). Automatically summarising web sites: is there a way around it? In *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM)*, (pp. 173–179). ACM.

Banerjee, S., Mitra, P., & Sugiyama, K. (2015). Multi-document abstractive summarization using ilp based multi-sentence compression In *Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, (pp. 1208–1214).

Cao, Z., Wei, F., Dong, L., Li, S., & Zhou, M. (2015, February). Ranking with Recursive Neural Networks and Its Application to Multi-Document Summarization. In *AAAI*, (pp. 2153–2159).

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297.

Delort, J. Y., Bouchon-Meunier, B., & Rifqi, M. (2003). Enhanced Web Document Summarization Using Hyperlinks. In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, (pp. 208–215). ACM.

Delort, J. Y. (2006). Identifying commented passages of documents using implicit hyperlinks. In *Proceedings of the Seventeenth Conference on Hypertext and Hypermedia*, (pp. 89–98). ACM.

Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479.

Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An Efficient Boosting Algorithm for Combining Preferences. *Journal of Machine Learning Research, 4(Nov)*, 933–969.

Gao, W., Li, P., & Darwish, K. (2012). Joint Topic Modeling for Event Summarization across News and Social Media Streams. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, (pp. 1173–1182). ACM.

Gong, Y., & Liu, X. (2001). Generic Text Summarization using Relevant Measure and Latent Semantic Analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 19–25). ACM.

Hong, K., & Nenkova, A. (2014). Improving the estimation of word importance for news multi-document summarization. In *The 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, (pp. 712–721). Association for Computational Linguistics.

Hu, M., Sun, A., & Lim, E. P. (2008). Comments-Oriented Document Summarization: Understanding Document with Readers' Feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, (pp. 291–298). ACM.

Hu, P., Sun, C., Wu, L., Ji, D. H., & Teng, C. (2011). Social Summarization via Automatically Discovered Social Context. In *IJCNLP*, (pp. 483–490).

Jones, K. S. (2007). Automatic summarising: The state of the art. *Information Processing & Management, 43(6)*, 1449–1481. Elsevier.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. In *Nature 401(6755)*, 788-791.

Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, (pp. 556–562).

Lee, J. H., Park, S., Ahn, C. M., & Kim, D. (2009). Automatic generic document summarization based on non-negative matrix factorization. In *Information Processing & Management, 45(1)*, 20-34. Elsevier.

Li, C., Wei, Z., Liu, Y., Jin, Y., & Huang, F. (2016). Using relevant public posts to enhance news article summarization. In *In Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, (pp. 557–566). Association for Computational Linguistics.

Lin, H. (2007). Projected gradient methods for nonnegative matrix factorization. In *Neural computation, 19(10)*, 2756–2779. MT Press.

Lin, H., & Bilmes, J. (2011). A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, (Volume 1, pp. 510–520). Association for Computational Linguistics.

Lin, C. Y., & Hovy, E. (2003). Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology* (Volume 1, pp. 71–78). Association for Computational Linguistics.

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research Development*, *2*(2), 159–165.

Nakov, P., Popova, A., & Mateev, P. (2001) Weight functions impact on lsa performance. In *Proceedings of EuroConference Recent Advances in Natural Langueage Processing (RANLP)*, (pp. 187–193).

Nallapati, R., Zhai, F., & Zhou, B. (2016) SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents. In *AAAI 2017*.

Nenkova, A. (2005). Automatic text summarization of newswire: lessons learned from the document understanding conference. In *AAAI*, (Vol.5, pp. 1436–1441).

Nenkova, A., & McKeown, K. (2011). Automatic summarization. In *Foundations and Trends in Information Retrieval, 5(23)*, 103–233).

Nguyen, M. T., Tran, V. C., Nguyen, X. H., & Nguyen, M. L. (2017). Utilizing User Posts to Enrich Web Document Summarization with Matrix Cofactorization. In *Proceedings of The Eight International Symposium on Information and Communication Technology (SoICT)*, (pp. 70–77). ACM.

Nguyen, M. T., Tran, V. D., Tran, C. X., & Nguyen, M. L. (2017b). Exploiting User-Generated Content to Enrich Web Document Summarization. In *International Journal on Artificial Intelligence Tools, 26(5)*, 1–26.

Nguyen, M. T., & Nguyen, M. L. (2017). Intra-relation or inter-relation?: Exploiting social information for Web document summarization. In *Expert Systems with Applications, 76*, 71–84.

Nguyen, M. T., Lai, V. D., Do, P. K., Tran, D. V., & Nguyen, M. L. (2016). VSoLSCSum: Building a Vietnamese Sentence-Comment Dataset for Social Context Summarization. In *Proceedings of the 12th Workshop on Asian Language Resources (ALR)*, (pp. 2409–2412). Association for Computational Linguistics.

Nguyen, M. T., Tran, V. D., Tran, C. X., & Nguyen, M. L. (2016). Learning to Summarize Web Documents using Social Information. In *Proceedings of ICTAI*, (pp. 619-626). IEEE.

Nguyen, M. T., Tran, C. X., Tran, D. V., & Nguyen, M. L. (2016). Solscsum: A linked sentence-comment dataset for social context summarization. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM)*, (pp. 2409–2412). ACM.

Nguyen, M. T., & Nguyen, M. L. (2016). SoRTESum: A Social Context Framework for Single-Document Summarization. In *European Conference on Information Retrieval (ECIR)*, (pp. 3–14). Springer International Publishing.

Mihalcea, R., & Tarau, P. (2004, July). TextRank: Bringing order into texts. Association for Computational Linguistics.

Park, S., Lee, J. H., Ahn, C. M., Hong, J. S., & Chun, S. J. (2006). Query based summarization using non-negative matrix factorization. In *Knowledge-Based Intelligent Information and Engineering Systems (KSE)*, (pp. 84–89). Springer Berlin/Heidelberg.

Ren, P., Chen, Z., Ren, Z., Wei, F., Ma, J., & Rijke, M. (2017). Leveraging contextual sentence relations for extractive summarization using a neural attention model. In *Proceedings of the 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, (pp. 95–104). ACM.

Shen, D., Sun, J. T., Li, H., Yang, Q., & Chen, Z. (2007). Document Summarization Using Conditional Random Fields. In *IJCAI*, (Vol.7, pp. 2862–2867).

Sun, J. T., Shen, D., Zeng, H. J., Yang, Q., Lu, Y., & Chen, Z. (2005). Webpage summarization using clickthrough data. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, (pp. 194–201). ACM.

Wang, D., Li, T., Zhu, S., & Ding, C. (2008). Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, (pp. 307–314). ACM.

Wei, Z., & Gao, W. (2014). Utilizing Microblogs for Automatic News Highlights Extraction. In *COLING*, (pp. 872–883). Association for Computational Linguistics.

Wei, Z., & Gao, W. (2015). Gibberish, Assistant, or Master?: Using Tweets Linking to News for Extractive Single-Document Summarization. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, (pp. 1003–1006). ACM.

Woodsend, K., & Lapata, M. (2010, July). Automatic generation of story highlights. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, (pp. 565–574). Association for Computational Linguistics.

Yang, Z., Cai, K., Tang, J., Zhang, L., Su, Z., & Li, J. (2011). Social Context Summarization. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 255–264). ACM.

Yeh, J. Y., Ke, H. R., Yang, W. P., & Meng, I. H. (2005). Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing & Management*, *41*(1), 75–95.