Summarizing Web Documents using Sequence Labeling with User-generated Content and Third-party Sources

Minh-Tien Nguyen[12], Duc-Vu Tran[1], Chien-Xuan Tran[1], and Minh-Le Nguyen[1]

[1] Japan Advanced Institute of Science and Technology (JAIST),
1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan.
[2] Hung Yen University of Technology and Education (UTEHY), Hung Yen, Vietnam.
{tiennm,vu.tran,chien-tran,nguyenml}@jaist.ac.jp

Abstract. This paper presents *SoCRFSum*, a summary model which integrates user-generated content as comments and third-party sources such as relevant articles of a Web document to generate a high-quality summarization. The summarization was formulated as a sequence labeling problem, which exploits the support of external information to model sentences and comments. After modeling, Conditional Random Fields were adopted for sentence selection. *SoCRFSum* was validated on a dataset collected from Yahoo News. Promising results indicate that by integrating the user-generated and third-party information, our method obtains improvements of ROUGE-scores over state-of-the-art baselines.

Keywords: Data Mining, Document Summarization, Social Context Summarization, Sequence Labeling.

1 Introduction

Online news providers, e.g. Yahoo News[3] interact with readers by providing a Web interface where readers can write their comments corresponding to an event collected from an original Web page. For example, after reading the Boston bombing event mentioned in an article, users can directly write their comments (user-generated content) on the interface of Yahoo News. In the meantime, by searching the title of the article using a search engine, we can retrieve relevant Web documents (third-party sources), which have two characteristics: (i) they have an implicit relation with the original document and (ii) they also include the event content in a variation. The user-generated content and third-party sources, one form of social information [1,5,22,9,14,26,24,17], cover important aspects of a Web document. This observation inspires a challenging summary task, which exploits the user-generated content and third-party sources of a Web document to support sentences for generating a summarization.

Extractive summarization methods usually formulate sentence selection as a binary classification task [11,21,26], in which they mark summary sentences by

[3] http://news.yahoo.com

a label. These methods, however, only consider internal document information, e.g. sentences while ignoring its social information. How to elegantly formulate sentence-comment relation and how to effectively generate high-quality summaries by exploiting the social information are challenging questions.

Social content summarization has been previously studied by various approaches based on different kinds of social information such as hyperlinks [1], click-though data [22], user-generated content [9,14,26,8,24,17]. Yang et al. [26] proposed a dual wing factor graph model for incorporating tweets into the summarization and used Support Vector Machines (SVM) and Conditional Random Fields (CRF) as preliminary steps in calculating the weight of edges for building the graph. Wei and Gao [24] used a learning to ranking approach with a set of features trained by RankBoost for news highlight extraction. In contrast, Gao et al. [8] proposed a cross-collection topic-aspect modeling (cc-TAM) as a preliminary step to generate a bipartite graph, which was used by a co-ranking method to select sentences and tweets for multi-document summarization. Wei and Gao [25] proposed a variation of LexRank, which used auxiliary tweets for building a heterogeneous graph random walk (HGRW) to summarize single documents. Nguyen and Nguyen [17,18] presented SoRTESum, a ranking method using a set of recognizing textual entailment (RTE) features for single-document summarization. These methods, however, exist two issues: (i) ignoring the support from third-party sources which can be seen as a global information and (ii) eliminating sequential aspect which is the nature of the summarization.

Our objective is to automatically extract summary sentences and comments of a Web document by incorporating its user-generated content and third-party sources. This paper makes four main contributions:

- It denotes the relation of sentences and comments in a mutual reinforcement fashion, which exploits both local and social information.
- It proposes sophisticated features which integrate the social information into the summary process.
- It conducts a careful investigation to evaluate feature contribution, which benefits the summarization in selecting appropriate features.
- It presents a unified summary framework which exploits user-generated content and third-party sources for producing the summarzation.

We next introduce our idea and data preparation for the summarization, then we describe the process of SoCRFSum. After generating the summary, we show compared results over baselines with discussions and deep analysis. We finish by drawing important conclusions.

2 Summarization with User-generated Content and Third-party Sources

2.1 Basic Idea

We formulate the summarization in the form of sequence labeling, which integrates the user-generated content and third-party sources of a Web document.

Such information represents supporting features, which help to enrich the information of each sentence or comment. For example, in Figure 1, when modeling a sentence, a set of comment features (the red line) and third-party features (the purple line) from relevant documents are exploited to support local features. Similarly, social features from sentences (the blue line) and third-party features are also utilized to support local features of each comment. In this view, we also consider information from sentences as the social features of comments. After modeling, a dual CRF-based [12] summarization model is trained to select sentences and comments as the summarization.

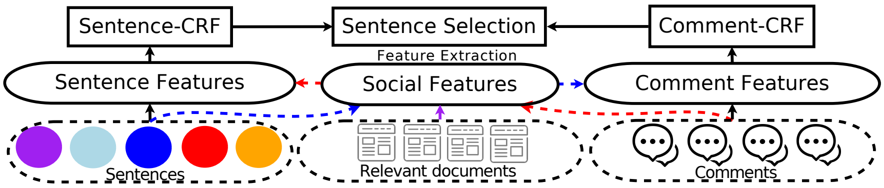


Fig. 1: The overview of SoCRFSum.

Our approach is different from [21,23,8,25,17,24,20] in two aspects: (i) exploiting both the information from users and relevant documents and (ii) formulating the summarization as a sequence labeling task.

2.2 Data Preparation

Since DUC (2001, 2002 and 2004)[4] lacks social information, we used a standard dataset for social context summarization named SoLSCSum from [19]. The SoLSCSum [19] is an English dataset collected from Yahoo News[5]. It contains 157 news articles along with 3,462 sentences, 5,858 gold-standard references, and 25,633 comments, which were manually annotated. To create third-party sources, we manually retrieved 1,570 related documents by selecting top 10 Web pages, which appear on the search result page of Google[6] by searching the title of each document. Unnecessary information, e.g. HTML tags was removed to obtain raw texts. We kept the order of the relevant documents.

2.3 Summarization by SoCRFSum

Basic Model with Local Information: As mentioned, our goal is to build a sequence model for the summarization. To do that, we followed the model with basis features stated in [21]. The basic features include sentence position, sentence length (the number of tokens with stopword removal), log-likelihood, thematic words, indicator words (count the number of indicator words using a dictionary), uppercase words, Cosine similarity with previous and next N ($N = 1, 2, 3$) sentences, LSA and HIT score. The remaining sections describe our new features used to improve the performance of the summary model.

[4] http://duc.nist.gov/data.html
[5] https://www.yahoo.com/news/
[6] https://www.google.com

New Local Features: We define a set of new features[7], which represent inherent information of each sentence. They capture the similarity of a sentence (or comment) with a title, informativeness, and topical covering aspect.

- Common Word: counts common words of the title and a sentence.
- Stop Word: counts the number of stop words in a sentence or comment.
- Local LDA Score: bases on a hypothesis that a summary sentence or comment usually contains salient information represented in the form of topics. We trained two topic models for sentences and comments by using Latent Dirichlet Allocation (LDA[8]) [2] ($k = 100$, $\alpha = \beta = 0.01$ with 1000 iterations). Given a document d with its comments, we formed sentences and comments into d_S and d_C, two smaller documents. For each d_S (or d_C), LDA infers to obtain top five topics, which have the highest topic distribution with d_S (or d_C). With each topic, we select top five topical words, which have the highest weights. As the result, each d_S (or d_C) has 25 topical words. Given a set of topical words $TW = \{w_1, ..., w_k\}$, Eq. (1) defines the local LDA score of a sentence:

$$\textit{local-lda-score}(s_i) = \frac{\sum_{j=1}^{k} weight(w_j)}{n} \text{ if } w_j \in s_i \tag{1}$$

  where: $weight()$ returns the word weight of w_j (normalized in [0, 1]) in s_i; n is the number of words in s_i after removing stop words.

Social Context Integration: We consider the user-generated content, e.g. comments and third-party sources, e.g. relevant news articles returned by a search engine as the social context of a Wed document d. In this view, the content of a document is not only mentioned in comments but also described in other Web pages from different news providers. We generated 13 new features, in which each feature covers the characteristic of a summary sentence or comment, that may match with the gold-standard references.

User-generated Features: cover semantic similarity, topic, and textual entailment aspect of a sentence-comment pair.

- Maximal W2V Score: This feature captures the generation behavior of readers, in which they generate comments based on sentences in a variation form. For example, a sentence and comment containing word *"police"* and *"cops"* can be considered to be similar. Eq. (2) defines this feature:

$$\textit{w2v-score}(s_i) = \max\left(\underset{j=1}{\overset{m}{sentSim}}(s_i, c_j)\right) \tag{2}$$

  where: m is the number of comments, $sentSim()$ returns semantic similarity of s_i and c_j and is calculated by Eq. (3):

$$sentSim(s_i, c_j) = \frac{\sum_{w_i}^{N_s} \sum_{w_j}^{N_c} w2v(w_i, w_j)}{N_s + N_c} \tag{3}$$

[7] We remove stopwords when modeling all features

[8] http://mallet.cs.umass.edu

where: N_s and N_c are the number of words in s_i and c_i; $w2vSim()$ returns the similarity of two words computed by *Word2Vec* [15].

- Auxiliary LDA Score: states that the content of a summary sentence should also appear in comments represented by topics. This feature is similar to *Local LDA Score* but topics and topical words were inferred from comments.
- Maximal Lexical Similarity: denotes the lexical similarity of a sentence and auxiliary comments, in which the content of a summary sentence should appear in several comments. We exploited similarity measures in [17] for modeling this feature. Eq. (4) presents the lexical similarity aspect:

$$rte\text{-}lex(s_i) = \max\left(\underset{j=1}{\overset{m}{lexSim}}(s_i, c_j)\right) \tag{4}$$

  where: m is the number of comments; $lexSim()$ returns the lexical similarity of s_i and c_j and is computed by Eq. (5):

$$lexSim(s_i, c_j) = \frac{1}{|F|}\sum_{n=1}^{|F|} f_n(s_i, c_j) \tag{5}$$

  where: F contains lexical features [17]; $f_n()$ is a similarity function computed by each n^{th} feature.
- Maximal Distance Similarity: This feature shares observation with the lexical feature but using distance aspect. It was calculated by the same mechanism in Eq. (4), but using distance features [17]. By adding two new similarity features, we cover the entailment aspect between a sentence and comment.

Third-party Features: An event in an original document is also mentioned in relevant Web documents from different news providers. We define the relation of the original and its related documents D as an implication because all the documents are created independently without the presence of social users. We only apply these features to the relevant documents instead of comments due to the implicit relation. The features capture social voting, social distance, and the appearance of frequent words of a sentence in the related documents D.

- Voting: bases on a hypothesis that the relevant documents should include salient terms in a summary sentence. Given a sentence s_i in the original document d, the voting in Eq. (6) counts Cosine similarity (greater than a threshold) with all sentences in the relevant documents.

$$n\_vote(s_i) = \frac{\sum_{j=1}^{m} cos(s_i, t_j)}{N_S} \tag{6}$$

  where: m is total sentences in the relevant documents and N_S is the number of sentences in d.
- Cluster Distance: states that a summary sentence should be close to clusters represented by the relevant documents, in which each of them is a cluster. Given a sentence s_i and a relevant document $rd_j \in D$ represented by a set of frequent terms, Eq. (7) denotes this feature.

$$c\text{-}dist(s_i, D) = \frac{\sum_{j=1}^{N_D} eucDist(s_i, rd_j)}{N_D} \tag{7}$$

where: $eucDist()$ returns Euclidean distance of s_i and rd_j using bag-of-words model, N_D is the number of relevant documents.

- Sentence-third-party Term Frequency: bases on a hypothesis that a sentence containing frequent words appearing in both of the original document and third-party sources is more important than other ones. Given a sentence s_i in d and N_{SD} is total sentences in D, Eq. (8) defines this feature.

$$stp\text{-}TF(s_i) = \frac{\sum_{j=1}^{|s_i|} TF(w_j) \times IDF(w_j)}{|s_i|} \tag{8}$$

$$TF(w_j) = \text{the frequency of } w_j \text{ in } d \tag{9}$$

$$IDF(w_j) = log(\frac{N_{SD}}{DF(w_j)}) \tag{10}$$

where: $DF(w_j)$ is the number of sentences in D containing w_j.

- Frequent-Terms Probability: The remaining features base on an assumption that the summarization should include the most frequent words if they appear frequently in the relevant documents. We first collected a set of frequent terms from the raw texts of D. If frequency of a term is greater than a certain threshold, it was considered to be frequent. Given a frequent term set $FT = \{w_1, ..., w_t\}$ and the original document d, we included three probability features: the average probability of frequent terms (aFrqScore), frequent term sum score (frqScore), and relative frequent term sum score (rFrqScore) [23]. Eqs. (11), (13) and (14) present these feature.

$$aFrqScore(s_i) = \frac{\sum_{w \in s_i} p(w)}{|w \in s_i|} \tag{11}$$

where: $w \in FT$.

$$p(w) = \frac{count(w)}{|w \in d|} \tag{12}$$

where: $count(w)$ is the frequency of w in d, and $|w \in d|$ is total number of frequent words in d.

$$frqScore(s_i) = \sum_{w \in s_i} p(w) \tag{13}$$

$$rFrqScore(s_i) = \frac{\sum_{w \in s_i} p(w)}{|s_i|} \tag{14}$$

Note that we also applied the features to model comments in the same mechanism. After modeling, two CRF-based models were separately trained for generating the summarization.

Summarization: We employed Viterbi[9] algorithm for decoding to generate the summarization. In decoding, if the number of sentences and comments labeled by 1 is larger than m, we select top m sentences and comments[10]; otherwise, we select all of them as the summarization.

[9] https://en.wikipedia.org/wiki/Viterbi_algorithm

[10] We do this because baselines also pick up top m sentences

3 Results and Discussion

3.1 Experimental Setup

Comments with fewer than five tokens were removed. 10-fold cross-validation was used with $m = 6$ as the same setting in [19]. Summaries were stemmed[11].

Word2Vec was derived by using SkipGram model[12] [15], dimension = 300, data from Google 1 billion words. Language models for learning to rank (L2R) baseline are uni-gram and bi-gram taken from KenLM[13]. The threshold of term frequency used for Eqs. (11), (13) and (14) is 0.65 when modeling sentences and 0.35 when modeling comments.

3.2 Baselines

- **SentenceLead:** chooses the first m sentences as the summarization [16].
- **LexRank[14]:** was proposed by [6]. We employed LexRank with tokenization and stemming[15].
- **SVM:** was proposed by [4] and used in [26,19]. We adopted SVM[16] by using RBF kernel with feature scaling in [-1, 1]; comments were weighted by 85% as the suggestion in [19] by using features in [26].
- **SoRTESum:** was proposed by [17] using a set of similarity features. This method contains two model: using inter information (SoRTESum Inter Wing) and dual wing information (SoRTESum Dual Wing).
- **CRF**: was used in [21] for single document summarization. We used basic features in [21] to train two summary models for sentences and comments.

3.3 Evaluation Method

Gold-standard references (labeled by 1) were used for evaluation by using F-1 ROUGE-N[17] (N=1, 2) [13].

3.4 Results

We report the summary performance of all methods in term of F-1 ROUGE-scores. The results in Tables 1 and 2 indicate that SoCRFSum clearly outperforms the baselines except for LexRank in ROUGE-1 on the comment side, i.e. 0.232 vs. 0.244. In sentence selection, our method slightly surpasses CRF with the basic features, e.g. 0.426 vs. 0.413 in ROUGE-1. It is understandable that the basic features in [21] are efficient to capture sequence aspect in documents.

[11] http://snowball.tartarus.org/algorithms/porter/stemmer.html
[12] https://code.google.com/p/word2vec/
[13] https://kheafield.com/code/kenlm/
[14] https://code.google.com/p/louie-nlp/source/browse/trunk/louie-ml/src/main/java/org/louie/ml/lexrank/?r=10
[15] http://nlp.stanford.edu/software/corenlp.shtml
[16] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/
[17] http://kavita-ganesan.com/content/rouge-2.0-documentation

On the other hand, our method obtains significant improvements compared to CRF in comment extraction, e.g. 0.232 vs. 0.074. This is because the sequence aspect does not exist in comments; therefore, basic features, e.g. Cosine (N=1,2,3) score are inefficient for covering summary comments. In this sense, our proposed features boot the summary performance.

Table 1: Sentence selection performance; * is supervised methods; **bold** is the best, *italic* is second best; methods with S use social information.

Method	**ROUGE-1**			**ROUGE-2**		
	Avg-P	**Avg-R**	**Avg-F**	**Avg-P**	**Avg-R**	**Avg-F**
Sentence Lead	0.749	0.242	0.365	0.635	0.216	0.322
LexRank	0.671	0.217	0.328	0.517	0.171	0.258
SVM*	0.712	0.185	0.293	0.573	0.151	0.239
SoRTESum Inter Wing (S)	0.705	0.239	0.357	0.582	0.201	0.298
SoRTESum Dual Wing (S)	0.721	0.242	0.362	0.593	0.203	0.302
CRF*	0.925	0.266	*0.413*	0.864	0.253	*0.391*
SoCRFSum* (S)	0.939	0.275	**0.426**	0.882	0.264	**0.407**

Table 2: Comment summarization, SentenceLead was not used.

Method	**ROUGE-1**			**ROUGE-2**		
	Avg-P	**Avg-R**	**Avg-F**	**Avg-P**	**Avg-R**	**Avg-F**
LexRank	0.541	0.157	**0.244**	0.360	0.087	*0.140*
SVM*	0.507	0.082	0.141	0.296	0.042	0.073
SoRTESum Inter Wing (S)	0.523	0.153	*0.237*	0.341	0.084	0.134
SoRTESum Dual Wing (S)	0.518	0.129	0.206	0.339	0.068	0.113
CRF*	0.639	0.039	0.074	0.539	0.033	0.062
SoCRFSum* (S)	0.870	0.134	0.232	0.767	0.121	**0.209**

In document summarization, SoCRFSum significantly outperforms SVM because our method exploits the sequence aspect and new features from additional sources. Our method achieves sufficiently improvements compared to other baselines because it exploits the support from both user-generated content and third-party sources. For example, there is a big gap between our approach and SoRTESum even though SoRTESum also integrates social information. The comparison is consistent in comment extraction. However, in some cases, our method is limited due to the sequence aspect in comments. For example, LexRank and SoRTESum Inter Wing slightly surpass our method in ROUGE-1, e.g. 0.237 vs. 0.232, but in ROUGE-2, our method is the best.

We compared our model to state-of-the-art methods reported in [19] for social context summarization. Table 3 indicates that our method performs significantly better than other methods except for ROUGE-1 in comment extraction. As mentioned, the lack of sequence aspect in comments challenges our approach. The results of L2R CCF and SVM Ranking suggest that formulating sentence extraction by learning to rank benefits the summarization. HRGW achieves competitive results due to the integration of social information. The results of cc-TAM are quite poor because it is used for multi-document summarization while the dataset in [19] is for single document summarization.

Table 3: Comparison results; *: supervised learning methods.

Method	**Document**		**Comment**	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
HGRW [25]	0.377	*0.321*	*0.248*	0.145
cc-TAM [8]	0.321	0.268	0.166	0.088
L2R* CCF [24]	0.363	*0.321*	0.217	0.111
SVM Ranking* [19]	*0.420*	0.317	**0.365**	*0.154*
SoCRFSum*	**0.426**	**0.407**	0.232	**0.209**

3.5 Feature Contribution

We investigated feature contribution by the minus ROUGE-scores of SoCRFSum using all and $n-1$ features (leave-one-out-test). Table 4 shows that in document summarization, most of the features positively affect our model. Our proposed features contribute to the model except for *stp-TF* and relative frequent term score (R-Frq-Score). In comment extraction, most of new features are positive except for *in title*, *#Stop words*, *local and aux LDA* compared to basic features, which most of them are negative. This observation explains the results in Tables 1 and 2. Interestingly, *#Stop words* is positive in sentence selection but is negative in comment extraction because comments are less formal than sentences. Also, *sentence position* and *in title* are inefficient for comments.

Table 4: Feature contribution, **bold**: new basic and *italic*: basic features.

Basic	**Document**		**Comment**		**User-generated**	**Document**		**Comment**		**Third-party**	**Document**		**Comment**	
Feature	**R-1**	**R-2**	**R-1**	**R-2**	**Feature**	**R-1**	**R-2**	**R-1**	**R-2**	**Feature**	**R-1**	**R-2**	**R-2**	**R-2**
Position	5x10-3	0.001	-0.034	-0.001	**In title**	9x10-3	0.003	-0.002	-0.002	Voting	0.002	0.003	8x10-3	0.003
Length	-0.003	-0.003	-0.005	-0.004	**#Stop words**	0.001	0.005	-0.015	-0.014	c-distance	0.001	0.002	0.035	0.028
Log-LH	-3x10-3	1x10-3	0.006	0.007	Local LDA	3x10-3	0.001	-2x10-3	-3x10-3	stp-TF	-0.001	-5x10-5	0.005	0.006
Them-word	7x10-3	7x10-3	-5x10-3	6x10-3	Aux LDA	3x10-3	0.001	-2x10-3	-3x10-3	A-Frq-Score	9x10-3	0.001	0.002	0.001
Upp-word	-8x10-3	-0.004	-0.013	-0.015	Max Cosine	6x10-5	-3x10-6	0.004	0.003	Frq-Score	1x10-3	-3x10-3	0.003	0.003
Cosine (N-1)	0.001	0.002	0.018	0.015	Max-dist RTE	3x10-3	0.001	0.007	0.007	R-Frq-Score	-3x10-3	7x10-3	0.004	0.003
Cosine (N-2)	-0.001	8x10-3	7x10-3	-6x10-3	Max-lex RTE	3x10-6	0.002	0.004	0.004	—	—	—	—	—
Cosine (N-3)	5x10-3	0.001	0.004	0.004	Max-aux W2V	3x10-3	0.001	0.007	0.007	*Cosine (N+2)*	0.002	0.002	0.004	0.002
Cosine (N+1)	-0.001	-7x10-3	0.009	0.006	—	—	—	—	—	*Cosine (N+3)*	-0.001	1x10-3	0.004	0.004
Ind-word	0.003	0.004	-0.010	-0.007	*LSA score*	0.001	0.002	-0.003	-3x10-3	*HIT score*	3x10-3	0.001	0.007	0.007

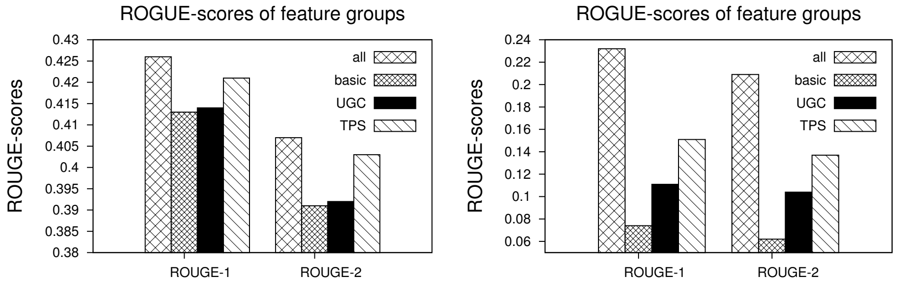
We also observed the contribution of feature groups by running our method with each separate group. Figure 2 indicates that in both document and comment extraction, the usage of all features benefits the summarization. In sentence selection, the model with the basic or user-generated features outputs similar performance, but in comment extraction, basic features are inefficient due to the lack of sequence aspect. Interestingly, in the both cases, the model with third-party features performs comparably to SoCRFSum with all features because many Web pages contain salient keywords in the summary sentences.

3.6 Summarization with L2R Methods

We observed the contribution of our features with learning to rank (L2R) methods by running three L2R methods: RankBoost [7], RankNet [3] implemented in RankLib[18], and SVM Ranking[19] [10] with default settings.

[18] http://people.cs.umass.edu/∼vdang/ranklib.html

[19] https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html



(a) Document summarization

(b) Comment summarization

Fig. 2: The contribution of feature groups; *UGC* denotes user-generated content features; *TPS* presents third-party-sources features.

Table 5: The performance of L2R methods with our features; methods with NF combine the old and new features.

System	**Document**		**Comment**	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
RankBoost	**0.417**	0.342	0.283	0.153
RankBoost (NF)	0.414	**0.344**	**0.285**	**0.158**
RankNet	0.388	0.317	0.219	0.120
RankNet (NF)	**0.389**	**0.318**	0.219	0.120
SVMRank	0.414	**0.345**	0.341	0.168
SVMRank (NF)	**0.421**	0.337	**0.379**	**0.175**

The results in Table 5 (NF means that an L2R method uses new features) show that our features contribute to improve the summary performance, especially with SVM Ranking. The improvement in comment extraction is larger than sentence selection. However, it is slightly increased compared to Tables 1 and 2. This is because our features may be appropriate for sequence labeling rather than L2R. The results in Tables 5 and 3 are consistent, in which L2R methods obtain competitive results. They point out that the summarization can also be presented in the form of L2R [23,24].

3.7 Error Analysis

In Table 6, SoCRFSum selects two correct sentences and comments (denoted by [+]) which clearly mention the event of Boston man shot by police. This is because summary sentences and comments contain important words *"Boston"*, *"police"* and *"arrest"*, which appear frequently in the comments and relevant documents. As a result, our features, e.g. max-rte lexical similarity, social voting, or sentence-third-party term frequency can efficiently capture these sentences. In addition, max-w2v score feature can represent a sentence-comment pair containing words *"police"* and *"arrest"* by using semantic similarity. This leads to the improvements in Tables 1 and 2.

On the other hand, non-summary sentences (denoted by [-]), e.g. S3 and C3 also including salient keywords challenge our method. For example, C3 contains

Table 6: A summary example of 121^{th} document. Three extracted sentences and comments are shown in stead of six ones due to space limitation.

Summary	
Sentences	**Comments**
[+]S1: The 26-year-old man, identified as Usaamah Rahim, brandished a knife and advanced on officers working with the Joint Terrorism Task Force who initially tried to retreat before opening fire, Boston Police Superintendent William Evans told reporters	[+]C1: "Boston Police and State Police made an arrest this evening in Everett"
[+]S2: "The FBI and the Boston Police did everything they could to get this individual to drop his knife," Evans said	[+]C2: This looks like the police are looking for an acceptable reason to shot and kill people
[-]S3: Law enforcement officials are gathered on a residential street in Everett, Massachusetts	[-]C3: It makes it sound like the police, armed with guns, were frightened nearly beyond control.

"police" and *"gun"*; therefore, our features are inefficient in this case. In addition, the similar sentence length of S3 and C3 also challenges our method. However, these sentences are still relevant to the event. SoCRFSum generates C1 which perfectly reflects the content of the document. C2 and C3 show the guess of readers after reading the document. These comments support sentences to provide a perspective viewpoint on the event.

4 Conclusion

This paper presents a summary model named *SoCRFSum* to address social context summarization. The model regards a document as a sequence of sentences and learns to generate a label sequence of sentences and comments. Our work is the first to combine both user-generated content from readers and relevant Web documents in a unified model, which operates in a mutual reinforcement fashion for modeling sentences or comments. We conclude that: (i) sequence labeling with the support of additional information improves the summarization over state-of-the-art baselines and (ii) our features and proposed model are efficient for summarizing single Web documents.

For future directions, an obvious next step is to examine how the model generalizes to other domains and text genres. More sophisticated features should be considered to address the sequence problem in comments.

Acknowledgment

This work was supported by JSPS KAKENHI Grant number JP15K16048, JSPS KAKENHI Grant Number JP15K12094, and JST CREST Grant Number JPMJCR1513, Japan.

References

1. Amitay, E., Paris, C.: Automatically summarising web sites: is there a way around it?. In: CIKM, pp. 173-179 (2000)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research **3**, 993-1022 (2003)

3. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G: Learning to rank using gradient descent. In: ICML, pp. 89-96 (2005)
4. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3), 273-297 (1995).
5. Delort, J.Y., Bouchon-Meunier, B., Rifqi, M.: Enhanced web document summarization using hyperlinks. In: Hypertext, pp. 208-215 (2003)
6. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. **22**, 457-479 (2004)
7. Freund, Y., Lyeryer, R.D., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research **4**, 933-969 (2003)
8. Gao, W., Li, P., Darwish, K.: Joint topic modeling for event summarization across news and social media streams. In: CIKM, pp. 1173-1182 (2012)
9. Hu, M., Sun, A., Lim, E.P.: Comments-oriented document summarization: Understanding document with readers' feedback. In: SIGIR, pp. 291-298 (2008)
10. Joachims, T.: Training linear svms in linear time. In: KDD, pp. 217-226 (2006)
11. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: SIGIR, pp. 68-73 (1995)
12. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML, 282-289 (2001)
13. Lin, C. Y., Hovy, E.: Automatic evaluation of summaries using n-gram cooccurrence statistics. In: HLT-NAACL Volume 1, pp. 71-78 (2003)
14. Lu, Y., Zhai, C., Sundaresan, N.: Rated aspect summarization of short comments. In: WWW, pp. 131-140 (2009)
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS, pp. 3111-3119 (2013)
16. Nenkova, A.: Automatic text summarization of newswire: lessons learned from the document understanding conference. In: AAAI, pp. 1436-1441 (2005)
17. Nguyen, M.T., Nguyen, M.L.: Sortesum: A social context framework for single-document summarization. In: ECIR, pp. 3-14 (2016)
18. Nguyen, M.T., Nguyen, M.L.: Intra-relation or inter-relation?: Exploiting social information for web document summarization. Expert Systems with Applications **76**, 71-84 (2017)
19. Nguyen, M.T., Tran, C.X., Tran, D.V., Nguyen, M.L.: Solscsum: A linked sentence-comment dataset for social context summarization. In: CIKM, pp. 2409-2412 (2016)
20. Nguyen, M.T., Tran, D.V., Tran, C.X., Nguyen, M.L.: Learning to summarize web documents using social information. In: ICTAI, pp. 619-626 (2016)
21. Shen, D., Sun, J.T., Li, H., Yang, Q., Chen, Z.: Document summarization using conditional random fields. In: IJCAI, pp. 2862-2867 (2007)
22. Sun, J.T., Shen, D., Zeng, H.J., Yang, Q., Lu, Y., Chen, Z.: Web-page summarization using clickthrough data. In: SIGIR, pp. 194-201 (2005)
23. Svore, K. M., Vanderwende, L., Burges, C. J.: Enhancing single-document summarization by combining ranknet and third-party sources. In: EMNLP-CoNLL, pp. 448-457 (2007)
24. Wei, Z., Gao, W.: Utilizing microblogs for automatic news highlights extraction. In: COLING, pp. 872-883 (2014)
25. Wei, Z., Gao, W.: Gibberish, assistant, or master?: Using tweets linking to news for extractive single-document summarization. In: SIGIR, pp. 1003-1006 (2015)
26. Yang, Z., Cai, K., Tang, J., Zhang, L., Su, Z., Li, J.: Social context summarization. In: SIGIR, pp. 255-264 (2011)