Learning to Summarize Web Documents using Social Information

Minh-Tien Nguyen*†, Duc-Vu Tran*, Chien-Xuan Tran* and Minh-Le Nguyen*
*Japan Advanced Institute of Science and Technology (JAIST),
1-1 Asahidai, Nomi, Ishikawa, 923-1292, JAPAN.
Email: {tiennm, vu.tran, chien-tran, nguyenml}@jaist.ac.jp
†Hung Yen University of Technology and Education (UTEHY), Hung Yen, Vietnam.

***Abstract*—This paper presents a method named *SoSVM-Rank*, which integrates the social information of a Web document to generate a high-quality summarization. In order to do that, the summarization was formulated as a learning to rank task, in which the order of a sentence or comment was determined by its informative information. The informative information was measured by a set of local and social features in which the social features were exploited to support the local ones when modeling a sentence or comment. To enrich information, new features were also proposed. After ranking, top *m* ranked sentences and comments were selected as the summarization. Our method was extensively evaluated on two datasets. Promising results indicate that: (1) by using new features, our method achieves improvements in both ROUGE-1 and ROUGE-2 of the summarization over state-of-the-art baselines and (2) integrating social information benefits the summarization.**

***Keywords*-Information Retrieval; Learning to Rank; Web Summarization; Social Context Summarization; NLP**

I. Introduction

Web 2.0 generation, e.g. Yahoo News[1] provides an interface where readers can write their comments regarding an event mentioned in a Web document. For example, after reading a Web document mentioning the Yemen capital bombing, readers can write their comments on the event on the Web interface. After writing these comments, other readers can immediately update the news content. These messages, one form of social information [1]–[4], not only reflect the content of a document and describe the facts of an event but also reveal the opinions of readers. This inspires a novel summarization task which utilizes the social information of a Web document to support sentences for generating the summarization.

Traditional extractive summarization methods select important sentences by using statistical or linguistic information in word/phrase or sentence level [4]–[8]. These methods, however, only consider inherent document information, e.g. sentences while ignoring its social information. How to elegantly integrate the social information into the summary process and effectively generate high-quality summaries are challenging questions.

Social context summarization (called the summarization) has received attention from researchers [2], [9]–[13]. Yang et al. proposed a dual wing factor graph model for incorporating tweets into the summarization [4]. The authors used Support Vector Machines (SVM) and Conditional Random Fields as preliminary steps in calculating the weight of edges for building the graph. Wei et al. used ranking approach with 35 features trained by RankBoost for news highlight extraction [3]. In contrast, Gao et al. proposed a cross-collection topic-aspect modeling (cc-TAM) as a preliminary step to generate a bipartite graph, which was used for co-ranking to select sentences and tweets for multi-document summarization [9]. Wei et al. proposed a variation of LexRank, which used auxiliary tweets in building a heterogenous graph random walk (HGRW) to summarize single documents [2]. Nguyen and Nguyen proposed SoRTESum, a ranking method using a set of recognizing textual entailment (RTE) features for single-document summarization [1]. However, some important aspects, e.g. sequence or semantic similarity are not completely considered in these methods.

The goal of this research is to automatically extract important sentences and representative comments of a Web document by incorporating its social information. This paper makes the following contributions:

- We propose new features to integrate social information into the summary process. The new features capture sequential, topical and semantic aspect.
- We carefully conduct an investigation to show the impact of each feature which benefits social context summarization in selecting appropriate features.
- We release an open-domain dataset[2] which contains news articles along with their comments. The standard extracted sentences and comments can be used to automatically evaluate the performance of summary systems in social context summarization. Our demo system can be also publicly accessed[3].

In order to achieve our goal, in next sections, we first introduce our idea of integrating social information into the summary process. From the idea, we present our model and new features to address the summarization. We also provide extensive experimental results and deep analysis to support our method and features.

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

[2] Download at: http://150.65.242.101:9292/yahoo-news.zip

[3] Demo system at: http://150.65.242.101:9293/?paper=ictai

2375-0197/16 $31.00 © 2016 IEEE
DOI 10.1109/ICTAI.2016.97

IEEE computer society

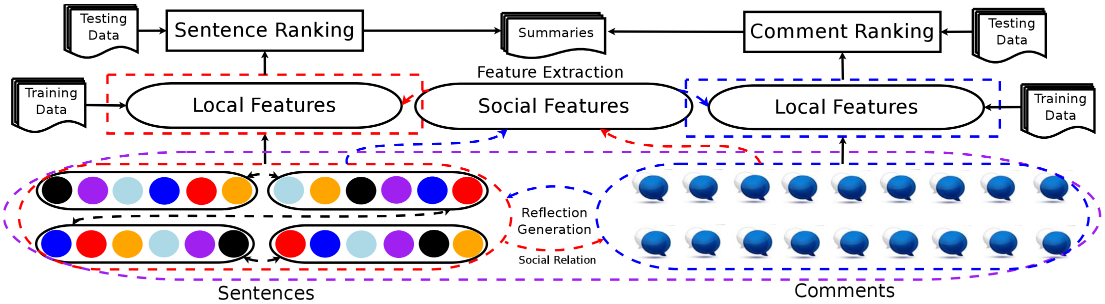


Figure 1: The overview of SoSVMRank, in which a document contains a set of sentences and comments.

II. Learning to Summarize using Social Information

A. Basic Idea

We formulate our problem in the form of learning to rank (L2R) [3], in which the summarization is generated by ranking sentences based on their informative information. This information can be measured by a set of local and social features in which the social features are utilized to support the local ones for enriching the informative information. In Figure 1, when modeling a sentence, a set of social features from comments were utilized to support local features, e.g. the red line. Similarly, a set of social features from sentences was also used to support local features in modeling a comment. In this view, sentences were also deemed as social information of comments. After modeling, two L2R models were separately trained on training data (sentences and comments) and applied to testing data. After ranking, top *m* ranked sentences and comments were selected as the summarization.

Our approach distinguishes with previous methods [1], [4]–[8] in two aspects: (1) formulating the summarization as a L2R task, (2) integrating social information. Our approach shares the idea of using learning to rank with [3]; however, adding new features, analyzing feature contribution and investigating the summarization by SVMRank are three key differences

B. Data Preparation

A new dataset was created by crawling up-to-date news articles from Yahoo News in May 2015. The dataset contains 157 open-domain articles along with 3,462 sentences, 5,858 extracted sentences as standard summaries and 25,633 comments. Two annotators were asked to annotate this dataset in two rounds. In the first round, each annotator read a complete article and selected sentences and comments (called instances) which mainly reflect the content of the web document. A selected instance would become a standard summary if the two annotators agree yes; otherwise, it is unimportant. The number of instances is no less than six for documents and 15 for comments; and no more than 35 in total. In the second round, the annotated data was cross-checked to show inter-annotator agreement. The inter-annotator agreement is 0.5845 calculated by Cohen's Kappa[4]. The dataset was created in 75 days. Data statistics are shown in Table I. The data is also shown in [14].

Table I: The data statistics; *s*: sentence and *c*: comment

# Documents	**Sentences**	**Standard Summaries**	**Comments**
157	3,462	5,858	25,633
# Tokens	78,634	116,845	375,836
# Average sentences/article	22.05	37.31	163.26
# Average tokens/article	500.85	744.23	2,393.85
# Average tokens/sentence	22.71	19.94	14.66
% positive examples	47.75	—	15.78
% Token overlapping	s/c: 13.26	—	c/s: 42.05
% Token overlapping (stop-word removal)	s/c: 8.90	—	c/s: 31.21

Table I shows that: (1) there exists common words or phrases between sentences and comments (the number in last two rows), (2) readers tend to use words or phrases appearing in sentences to create their comments (31.21% of word overlapping of comments on sentences). The data observation and literature review suggested four hypotheses: representation, reflection, generation, and common topic.

C. Learning to Summarize with SVMRank

To train the L2R model, we propose to use SVMRank [15], one state-of-the-art learning to rank methods for information retrieval [16]. SVMRank applies the characteristics of SVM to perform pairwise classification. Given n training queries $\{q_i\}_{i=1}^{n}$, their associated document pairs $(x_u^{(i)}, x_v^{(i)})$ and the corresponding ground truth label $y_{(u,v)}^{(i)}$, SVMRank optimizes an objective function shown in Eq. (1):

$$\min \frac{1}{2}\|w\|^2 + \lambda \sum_{i=1}^{n} \sum_{u,v:y_{u,v}^{(i)}} \xi_{u,v}^{(i)} \tag{1}$$

$$\text{s.t. } w^T(x_u^i - x_v^{(i)}) \geqslant 1 - \xi_{u,v}^{(i)}, \text{ if } y_{u,v}^{(i)} = 1 \tag{2}$$

$$\xi_{u,v}^{(i)} \geqslant 0,\ i = 1, ..., n \tag{3}$$

[4] http://graphpad.com/quickcalcs/kappa1.cfm

where: $f(x) = w^T x$ is a linear scoring function, (x_u, x_v) is a pairwise and $\xi^{(i)}_{u,v}$ is the loss. The document pair-wise is sentence-sentence or comment-comment. The pair-wise order was determined by a salient score between a sentence or comment with the ground-truth summary sentences [3]. Given $E = \{e_1, e_2, ..., e_k\}$ is a set of extracted standard sentences, the salient score of a sentence or comment was computed by Eq. (4)

$$score(I_i) = max\{RF(I_i, e_j)\},\ j \in \{1, k\} \qquad (4)$$

where: $RF()$ returns ROUGE-1 F-score between I_i and e_j.

D. Feature Extraction

We pushed social context summarization by proposing new features which are shown in Table II (the basic features can be seen in [3]).

Table II: Our proposed features; *italic* is a local feature

Local Features	Social Features
Sentence length (for sentence)	Maximal Word2Vec score
Sentence length before	Auxiliary LDA score
Sentence length after	Maximal-lexical RTE similarity
Cosine similarity before	Maximal-distance RTE similarity
Cosine similarity after	—
Local LDA score	*Function words*

1) Local Features: capture the inherent characteristics of a sentence and comment. By proposing new features, four important aspects of an individual instance were considered: length, sequence, topic covering and meaningless words.

- **Sentence length**: is the number of words in sentence s_i (no stop words).
- **Sentence length before**: is the number of words in sentence s_{i-1} giving s_i. The value is 0 if s_i is the first sentence in a document.
- **Sentence length after**: is the number of words in s_{i+1} giving s_i. The value is 0 if s_i is the last sentence in a document.
- **Cosine similarity before**: Given sentence s_{i-1} and s_i denoted by vector $\overrightarrow{x}$ and $\overrightarrow{y}$ using bag-of-words model, Cosine similarity was defined in Eq. (5):

$$cos(\overrightarrow{x}, \overrightarrow{y}) = \frac{\overrightarrow{x} . \overrightarrow{y}}{\parallel \overrightarrow{x} \parallel . \parallel \overrightarrow{y} \parallel} \qquad (5)$$

  where: x_i and y_i are the frequency of each word in s_i and s_{i-1}; $\overrightarrow{x}$ and $\overrightarrow{y}$ are two the same size vectors. The cosine value is 0 if s_i is the first sentence.
- **Cosine similarity after**: calculates the similarity of s_i and s_{i+1}. The value is 0 if s_i is the last sentence. By using sentence length and Cosine similarity before and after, the sequence aspect of the summarization was considered because a document is regarded as a sequence of sentences and selecting a sentence depends on its context.
- **Local LDA score**: captures the topical aspect of a document. This feature was calculated in two steps: training and inference. In the training step, two Latent Dirichlet Allocation (LDA) [17] models were trained on documents and comments to obtain document and word distribution over topics. In the inference step, given a document d along with top k ($k = 5$) closest topics inferred from LDA models, each topic has top $m = 5$ topical words, the local LDA score was calculated by Eq. (6):

$$local\text{-}lda\text{-}score(s_i) = \sum_{j=1}^{n} wordScore(w_j) \qquad (6)$$

  where: $wordScore()$ returns the word score of a word w_i (e.g. 0.45) in sentence s_i over top five topics generated by the inference; n is the number of words in s_i after removing stopwords, in which i is the index i^{th} of the sentence
- **Function words**: An informative sentence should contain content words rather than function ones, e.g. stop words. Let len_{org} to be the length of original sentence and len_{rmw} is the length of the sentence after removing stop words, this feature was computed by Eq. (7).

$$stop\text{-}word\text{-}count(s_i) = len_{org}(s_i) - len_{rmw}(s_i) \qquad (7)$$

2) Social Features: capture important information from the social information of a Web document in three aspects: semantic similarity, topical covering and entailment.

- **Auxiliary w2v score**: represents the semantic similarity between a sentence s_i and auxiliary comments. By using this feature, we attempted to cover sentences and comments which have a little word overlapping. For example, two sentences containing word *"police"* and *"cops"* can be considered to be similar. Given a sentence s_i and comment c_j, the semantic similarity feature was defined in Eq. (8):

$$w2v\text{-}score(s_i) = \max\{sentSim(s_i, c_j)\}\ j \in \{1, m\} \qquad (8)$$

  where: m is the number of comments, $sentSim()$ returns the semantic similarity of s_i and c_j and was calculated by Eq. (9):

$$sentSim(s_i, c_j) = \sum_{w_i}^{N_s} \sum_{w_j}^{N_c} w2vSim(w_i, w_j) \qquad (9)$$

  where: N_s and N_c are the number of words in s_i and c_i after removing stop words; $w2vSim()$ returns the semantic similarity between two words and was computed by *Word2Vec* [18].
- **Auxiliary LDA score**: is similar with *Local LDA score* but topics and important words were inferred from comments. This feature shows that an important sentence should also cover topics appearing in comments.
- **RTE lexical similarity**: denotes the lexical similarity of a sentence s_i and its auxiliary comments. We developed

the idea from [1], [19] by using features in RTE task for calculating the similarity between two texts. Intuitively, an important sentence can be deemed to be entailed with several representative comments. RTE lexical feature was defined in Eq. (11):

$$\textit{rte-lex}(s_i) = \max\{rteLexSim(s_i, c_j)\}\ j \in \{1, m\} \tag{10}$$

where: m is the number of comments; $rteLexSim()$ returns the lexical similarity of s_i and c_j and was computed by Eq. (11):

$$rteLexSim(s_i, c_j) = \frac{1}{F}\sum_{n=1}^{F} f_n(s_i, c_j) \tag{11}$$

where: F contains five lexical features [1], [19]; $f_n()$ is a similarity function computed by each n^{th} feature.

- **RTE distance similarity**: was calculated by the same mechanism with *RTE lexical* but using nine distance features [1], [19]. By proposing two new RTE features, the entailment aspect between a sentence and comment was also covered. Note that our proposed features were also used in the same mechanism for modeling comments.

E. Summarization

After ranking, top *m* ranked sentences and comments were selected as the summarization.

III. Results and Discussion

A. Experimental Setup

Comments with fewer than five tokens were removed. 10-fold cross validation with $m = 6$ (less than 30% of average sentences per document, see Table I) was used; stop words and links were removed; and summary instances were also stemmed[5] [20].

SVM[6] [21] was selected for the classification. SVMRank[7] was used to train the our model. RankLib[8] was used to train a learning to rank model. LDA[9] was used to generate topical words. *Word2Vec* was obtained by SkipGram model[10] [18], dimension = 300, data from Google 1 billion words. Uni-gram and bi-gram taken from KenLM[11] were used as language models for learning to rank (L2R).

B. Baselines

SoSVMRank was compared to the following baselines:

- **Random**: selects sentences and comments randomly.
- **SentenceLead:** chooses the first *x* sentences as the summarization [22].
- **SociVote**: selects sentences based on Cosine voting from comments [2]; the threshold = 0.65.
- **LexRank:** algorithm[12] [23]; tokenization and stemming[13] were used.
- **cc-TAM**: built a cross-collection topic-aspect modeling to generate a bipartite graph for co-ranking [9].
- **HGRW**: is a variation of LexRank named Heterogeneous Graph Random Walk [2]; the threshold was 0.7.
- **L2R**: was utilized by [3] using RankBoost [24] with 300 iterations and ERR metric score. Unnecessary features i.e., hashtags, URLs and quality depend were ignored. L2R has two methods: local sentence and comment features (L2R LSF-LCF), and all basic features (L2R CCF).
- **SVM:** RBF kernel was used with scaling in [-1, 1] (imbalance data on the comment side Table I), positive examples were weighted by 85%.
- **RTE One Wing:** uses one wing (document or comment/tweet) to calculate the RTE score.
- **SoRTESum:** was proposed by [1] using a set of RTE similarity features. It has two methods: SoRTESum Inter Wing and SoRTESum Dual Wing.
- **SVMRank:** uses the basic features in [3], without the new features with $C = 3.0$ and *linear* kernel.

C. Evaluation Method

F-1 ROUGE-N[14] (N=1, 2) [25] with stemming and stopword removal.

D. Experimental Results

Results in Table III show that our method obtains significant improvements from 0.5% to 15.3% and from 0.6% to 26.8% of ROUGE-1 in document and comment summarization over the baselines. This shows the efficiency of our method and proposed features and supports our hypotheses stated in Section II-B.

Our method outperforms SVMRank with the basic features in both ROUGE-1 and ROUGE-2 of document and comment summarization. SVMRank represents the relation of a sentence-comment pair by using local and cross features; however, it eliminates semantic similarity and sequential aspect. In contrast, our method resolves these issues by integrating new features leading improvements compared to SVMRank.

LexRank and HGRW achieve competitive results in ROUGE-2 of comment summarization (0.140 and 0.145) even they are unsupervised methods. This is because the number of word overlapping of comments per sentences is 31.21% (Table I); therefore, LexRank and HGRW tend to select comments including common words. However,

[5] http://snowball.tartarus.org/algorithms/porter/stemmer.html

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

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

[8] https://people.cs.umass.edu/∼vdang/ranklib.html

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

[10] https://code.google.com/p/word2vec/

[11] https://kheafield.com/code/kenlm/

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

[13] http://nlp.stanford.edu/software/corenlp.shtml

[14] http://kavita-ganesan.com/content/rouge-2.0-documentation

Table III: Summary performance on Yahoo News dataset; * is supervised method; **bold** is the best value; *italic* is the second best; SentenceLead was not used in summarizing comments. Methods with *S* use social information.

System	Document						Comment					
	ROUGE-1			ROUGE-2			ROUGE-1			ROUGE-2		
	P	R	F-1	P	R	F-1	P	R	F-1	P	R	F-1
Random	0.639	0.173	0.272	0.473	0.128	0.201	0.441	0.058	0.103	0.217	0.025	0.045
SentenceLead	0.749	0.242	0.365	0.635	0.216	*0.322*	—	—	—	—	—	—
SociVote (S)	0.72	0.213	0.328	0.589	0.18	0.276	0.549	0.09	0.155	0.385	0.054	0.095
LexRank	0.671	0.217	0.328	0.517	0.171	0.257	0.541	0.157	0.244	0.36	0.087	0.140
HGRW (S)	0.724	0.255	0.377	0.604	0.218	0.320	0.548	0.16	0.248	0.369	0.09	0.145
cc-TAM (S)	0.715	0.207	0.321	0.571	0.175	0.268	0.554	0.098	0.166	0.275	0.052	0.088
L2R* LSF-LCF (S)	0.742	0.232	0.353	0.619	0.204	0.307	0.445	0.133	0.205	0.262	0.06	0.098
L2R* CCF (S)	0.699	0.245	0.363	0.606	0.218	0.321	0.47	0.141	0.217	0.299	0.068	0.111
SVM*	0.712	0.185	0.293	0.573	0.151	0.239	0.507	0.082	0.141	0.296	0.042	0.074
RTE-One Wing	0.706	0.246	0.364	0.582	0.206	0.304	0.514	0.15	0.232	0.338	0.079	0.128
SoRTESum Inter Wing (S)	0.705	0.239	0.357	0.582	0.201	0.299	0.523	0.153	0.237	0.341	0.084	0.135
SoRTESum Dual Wing (S)	0.721	0.242	0.362	0.593	0.203	0.302	0.518	0.129	0.206	0.339	0.068	0.113
SVMRank* (S)	0.782	0.287	*0.420*	0.622	0.213	*0.317*	0.598	0.263	*0.365*	0.321	0.101	*0.154*
SoSVMRank* (S)	0.784	0.291	**0.425**	0.613	0.219	**0.323**	0.614	0.265	**0.371**	0.326	0.104	**0.158**

LexRank and HGRW are sensitive to the noise of data. This is because they used *IDF-modified-cosine similarity*; hence, these methods need a large corpus to calculate term frequency (TF) and inverse document frequency (IDF) based on bag-of-words model. This challenges the summarization [23]. The results of LexRank in Table IV supports this conclusion.

Table III also indicates that L2R with cross features (L2R CCF) and SoRTESum are competitive methods because they integrate social information. Sentence Lead is a strong baseline [22] because it formulates the summarization by taking some first sentences (both Tables III and IV). SVM obtains poor results due to the low quality of classification (0.6 of F-1 in sentence classification and 0.74 of F-1 in comment classification). This suggests a careful feature combination should be considered. cc-TAM achieves poor results because this method was proposed for multi-document summarization whereas our dataset was collected for single-document summarization. SociVote obtains reasonable results even it is a simple method. This shows the contribution of social information for the summarization.

Our method was extensively evaluated on a new highlight dataset taken from USAToday and CNN [3] (*m*=4 because each document has 3-4 highlights; 5-fold cross validation). The dataset contains 121 events, 455 highlights and 78.419 tweets with no labels. Near-duplicate tweets were removed [26] using Simpson formula.

Table IV indicates that SVMRank is the best in document summarization; however, in comment summarization, in ROUGE-1, our method is the best (0.233 vs. 0.226). We guess that conflict might appear when combining all features. Our method outperforms SVMRank in tweet summarization showing that our features may be appropriate for tweets. This also suggests that feature combination should be considered. The results of tweet summarization are competitive with sentence summarization because readers usually use a sentence or long phrases to create their tweets.

Table IV: Summary results on USAToday and CNN dataset

System	Document		Tweet	
	R-1	R-2	R-1	R-2
Random	0.167	0.037	0.156	0.059
Sentence Lead	0.249	*0.096*	—	—
LexRank	0.183	0.045	0.154	0.056
L2R* CCF	0.241	0.069	*0.230*	**0.076**
RTE-Sum one wing (S)	0.202	0.072	0.191	0.067
SoRTESum Inter Wing (S)	0.255	**0.098**	0.201	0.068
SoRTESum Dual Wing (S)	0.254	*0.096*	0.209	*0.074*
SVMRank* (S)	**0.279**	0.079	0.226	0.063
SoSVMRank* (S)	*0.261*	0.062	**0.233**	*0.071*

E. Feature Contribution

The contribution of our new features was investigated by averaging feature weight generated from the model in each fold. Results in Table V indicate that in document summarization, local features, e.g. sentence length, cosine similarity with the next and the previous sentence, function word positively contribute our model while sentence length of the next and previous sentence, local topical score negatively affect the model. The social features, e.g. Word2Vec score and RTE score also play an important role in our model whereas auxiliary topical score is negative. This trend is similar in comment summarization. From this, we conclude that sequence aspect in the form of Cosine similarity, semantic similarity, and RTE positively influence the model. Interestingly, function word is positive in document summarization but is negative in comment summarization because stop words appear frequently in comments.

A further investigation of feature group contribution was also conducted. Each feature group was removed while keeping the other. The F-score ratio was computed by the F-score minus of the model using all features and the model using one group. Note that the feature group contains both the basic and new features. Figure 2 shows that both local and social features contribute the summarization. Local information contributes a big influence in sentence and comment

Table V: Our feature contribution in document and comment summarization; *italic* is local feature.

Document				Comment			
Local Features	**Weight**	**Social Features**	**Weight**	**Local Features**	**Weight**	**Social Features**	**Weight**
Sent-length	1.533	Max-W2V score	1.768	—	—	Max-W2V score	0.085
Sent-length before	-0.283	Aux-LDA score	-0.101	Sent-length before	-0.655	Aux-LDA score	-0.153
Sent-length after	-0.251	Max-RTE-lex similarity	0.024	Sent-length after	-0.317	Max-RTE-lex similarity	2.197
Cos-sim before	1.039	Max-RTE-dis similarity	0.216	Cos-sim before	0.475	Max-RTE-dis similarity	0.270
Cos-sim after	1.044	—	—	Cos-sim after	0.217	—	—
Local LDA score	-0.267	*Function word*	0.489	Local LDA score	-0.415	*Function word*	-1.155

summarization. In comment summarization, the contribution of social features increases. Generally, the social features support the local features in generating summaries.

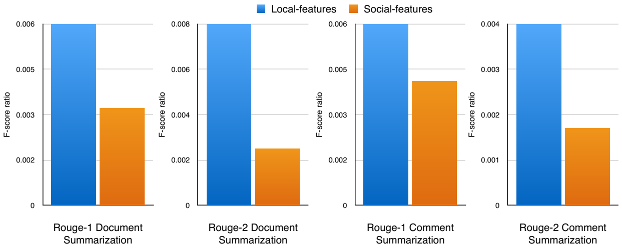


Figure 2: The contribution of two feature groups

The contribution of our new features was also investigated on RankBoost (L2R CCF in Table III) and SVMRank with two feature groups: the basic (without our features) and all features (including our new features). Results in Table VI show that our features contribute to improving the performance of RankBoost (1.6%, 0.379 vs. 0.363) in ROUGE-1 of document summarization. However, with SVMRank, the improvement is 0.5% (0.425 vs. 0.420). This trend is similar in ROUGE-1 of comment summarization. This indicates that our feature may be appropriate for RankBoost.

Table VI: Our feature contribution on two methods

Group	**RankBoost**				**SVMRank**			
	Document		**Comment**		**Document**		**Comment**	
	R-1	**R-2**	**R-1**	**R-2**	**R-1**	**R-2**	**R-1**	**R-2**
Basic	0.363	0.321	0.217	0.111	0.420	0.317	0.365	0.154
All	0.379	0.324	0.252	0.128	0.425	0.323	0.371	0.158

F. Summary Performance with L2R Methods

We run four learning to rank methods: RankBoost [24] (iteration = 300, metric is ERR10), Coordinate Ascent (random restart = 2, iteration = 25, tolerance = 0.001 with non-regularization), SVMRank, and SoSVMRank [15] ($C = 3$ with linear kernel) with all features (the basic and our new features) and tested the methods on our dataset. Results from Table VII show that SoSVMRank is the best. SVMRank, RankBoost and Coordinate Ascent are competitive methods. This is because SVMRank inherits nice properties of SVM. For example, it can create correct margins for classification based on the help of margin maximization. In training, this property helps SVMRank to avoid the overfitting problem which can appear in other methods, e.g. RankBoost or Coordinate Ascent. The results of L2R using RankBoost in Table III support this statement.

Table VII: The performance of learning to rank methods

L2R Method	**Document**		**Comment**	
	R-1 F-1	**R-2 F-1**	**R-1 F-1**	**R-2 F-1**
RankBoost	0.379	**0.324**	0.252	0.128
Coordinate Ascent	0.398	0.294	0.354	0.153
SVMRank	0.420	0.317	0.365	0.154
SoSVMRank	**0.425**	**0.323**	**0.371**	**0.158**

G. Sentence Position Observation

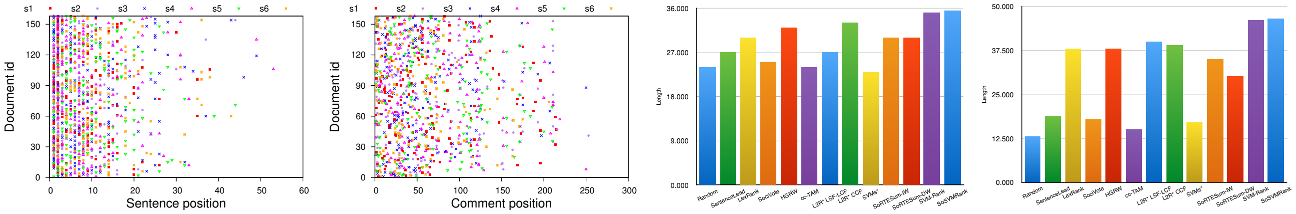
The position distribution of extracted sentences generated from our method was investigated. From Figures 3a and 3b, we observe that most important sentences are located within the first 15 sentences on the document side and 100 comments on the comment side. There are also some outlier points, e.g. 54^{th} in Figure 3a and 252^{th} in Figure 3b because some documents contains the larger number of sentences and comments. Considering the data observation in Table I, we conclude that the density distribution of comments is scattered because sequence aspect does not explicitly exist on the comment side.

H. Sentence Length Observation

Figures 3c and 3d indicate that longer sentences belong to the best models i.e., Sentence Lead and L2R CCF. SoSVMRank generates the longest sentences, i.e. 35.5 on document summarization and 46.5 on comment summarization supporting results in Table III. Other models obtain a similar sentence length, e.g. LexRank, SoRTESum and L2R while poor models achieve shorter length, i.e. Random, cc-TAM, SVM. This trend is similar in Figure 3d. This explains that the performance of Random, cc-TAM, and SVM is poor in generating summaries (see in Table III).

I. Hypothesis Analysis

A running example using SoRTESum Dual Wing [1] was conducted on USAToday and CNN dataset. Table VIII and Figure 4 indicate that S_1 and T_2 are important sentences which receive higher score whereas irrelevant sentences obtain lower score. This observation supports *representation* and *reflection* hypothesis. In addition, we observe that sentences and tweets share common words, e.g. Tamerlan, bombing, dead supporting *generation* and *common topic* hypothesis stated in Section II-B.



(a) On the document (b) On the comment (c) On the document (d) On the comment

Figure 3: Sentence and comment position distribution

Table VIII: Two sentences and tweets taken from USAToday and CNN dataset [3], S_1 and T_2 are summary sentences.

Sentences	Tweets
[S1] Police have identified Tamerlan Tsarnaev as the dead Boston bombing suspect	[T1] Who is Tamerlan Tsarnaev, 26, the man ID&#39 as the dead #BostonBombing
[S2] The brothers had been living together on Norfolk Street in Cambridge	[T2] Before his death Tamerlan Tsarnaev called an uncle andasked for his forgiveness. Said he is married and has a baby

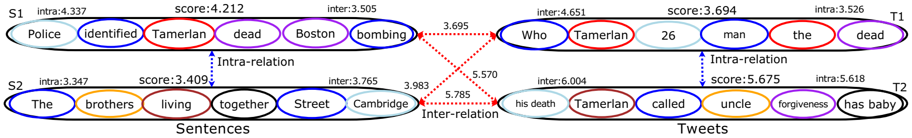


Figure 4: A running example from Table VIII implemented from [1].

J. Error Analysis

A further analysis of extracted sentences generated from our model was intensively conducted. In Table IX, our model yields correct sentences (denoted by [+]) which mention the death of Usaamah Rahim at Boston shooting event and the opinions of readers on this event. In document summarization, by using supports from comments, our model selects four correct sentences. This is because these sentences contain important information, i.e. the arrest of Boston Police and the description of Evans in the arrest mentioned in the document and its comments. As the result, our features can efficiently capture the informative information in each sentence. However, it also picks up two incorrect ones (S_2 and S_4, denoted by [-]) because they have a similar length with the correct ones and also contain important information. This challenges our model and shows that our proposed features are inefficient in some cases. However, S_2 and S_4 are still relevant to the Boston shooting event. Table IX also indicates that our proposed features, i.e. word2vec similarity and RTE similarity contribute the summarization.

In comment summarization, we found that candidate comments are long sentences and also share important phrases, e.g. *"drop the knife"*, *"cops"* and *"Boston"* with sentences. As the result, by using our features, the information from sentences benefits comment selection. However, our model also yields an incorrect comment (C_4) because it also has a similar sentence length. Extracted comments also show that they contain the opinions of readers (C_1 and C_5) and suggest solutions (C_2 and C_3). Interestingly, C_6 provides a new information of the arrest which was not mentioned in the document.

IV. Conclusion

This paper introduces SoSVMRank to address social context summarization. For this purpose, we formulate the summarization in the form of learning to rank, in which sentences and comments are arranged based on their informative information. To measure the informative information, we propose new features covering topical, sequential and semantic similarity aspect. Our features are used with the basic features to train the learning to rank model. After ranking, top *m* ranked sentences and comments are selected as summaries. For evaluation, we also release an open-domain dataset used to automatically evaluate summary systems in this task. This paper concludes that formulating sentence extraction in the form of learning to rank benefits the summarization. Promising results of the extensive evaluation indicate that our features are efficient for single-document summarization.

For future directions, other features, e.g. rhetorical relations between sentences or tree edit distance of the RTE task should be considered and integrated into the model. The summarization should be represented in a deeper semantic model, e.g. CNN or LSTM. Finally, human evaluation should be also considered to ensure the quality of the summarization.

Acknowledgment

We would like to thank John Blake for useful discussions and insightful comments on earlier drafts; Wei Gao and Peng Li for sharing the code of [9]. This work was supported by JSPS KAKENHI Grant number 3050941, JSPS KAKENHI Grant Number JP15K12094, and CREST, JST.

Table IX: A running example extracted from our method. The document is 121^{th}.

<table>
<tr><th colspan="2">Standard data</th></tr>
<tr><td colspan="2">[1] Either those cops weren't switched on enough to grasp the scope of the threat or Boston PD needs to review their procedures for addressing these types of threats.
[2] Law enforcement officers in Boston shot dead a man on Tuesday who came at them with a large knife when they tried to question him as part of a terrorism-related investigation, authorities said, describing him as a "threat."
[3] 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.
[4] "The FBI and the Boston Police did everything they could to get this individual to drop his knife," Evans said.
[5] If I had been one of the police officers I would have whispered 3 times "drop the knife" then quickly fired several shots at his sternum.</td></tr>
<tr><th colspan="2">Summary</th></tr>
<tr><th>Sentences</th><th>Comments</th></tr>
<tr><td>[+]S1: Law enforcement officers in Boston shot dead a man on Tuesday who came at them with a large knife when they tried to question him as part of a terrorism-related investigation, authorities said, describing him as a "threat."</td><td>[+]C1: "Fear for your life" is exactly like a "sincerely held belief", there's absolutely nothing to weigh and no measurement possible to make such a determination.</td></tr>
<tr><td>[-]S2: Boston Police said in a statement on their website that "as part of this ongoing investigation, Boston Police and State Police made an arrest this evening in Everett".</td><td>[+]C2: If I had been one of the police officers I would have whispered 3 times "drop the knife" then quickly fired several shots at his sternum.</td></tr>
<tr><td>[+]S3: 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.</td><td>[+]C3: Either those cops weren't switched on enough to grasp the scope of the threat or Boston PD needs to review their procedures for addressing these types of threats.</td></tr>
<tr><td>[-]S4: Evans said officers had approached the man in a strip-mall parking lot without weapons drawn and opened fire only after he repeatedly advanced on them, leaving them in fear for their lives.</td><td>[-]C4: Lawyers in a Union, lawyers in politics, they have made these unqualified sayings up, and its time to make them use more defined terms and refuse to accept escape path words that mean absolutely nothing.</td></tr>
<tr><td>[+]S5: A man who identified himself on Twitter as Rahim's brother said the family was shocked by the shooting.</td><td>[+]C5: Disturbed by the fact that they "didn't expect a reaction like this" and that they first retreated from this threat to themselves and others.</td></tr>
<tr><td>[+]S6: "The FBI and the Boston Police did everything they could to get this individual to drop his knife," Evans said.</td><td>[+]C6: Yet his Iman brother was already claiming he was shot in the back with this hands in the air.</td></tr>
</table>

REFERENCES

[1] M.-T. Nguyen and M.-L. Nguyen, "Sortesum: A social context framework for single-document summarization," in *ECIR: 3-14*, 2016.

[2] Z. Wei and W. Gao, "Gibberish, assistant, or master?: Using tweets linking to news for extractive single-document summarization," in *SIGIR: 1003-1006*, 2015.

[3] ——, "Utilizing microblogs for automatic news highlights extraction," in *COLING: 872-883*, 2014.

[4] Z. Yang, K. Cai, J. Tang, L. Zhang, Z. Su, and J. Li, "Social context summarization," in *SIGIR: 255-264*, 2011.

[5] J. Kupiec, J. O. Pedersen, and F. Chen, "A trainable document summarizer," in *SIGIR: 68-73*, 1995.

[6] M. Osborne, "Using maximum entropy for sentence extraction," in *ACL Workshop on Automatic Summarization: 1-8*, 2002.

[7] J.-Y. Yeh, H.-R. Ke, W.-P. Yang, and I.-H. Meng, "Text summarization using a trainable summarizer and latent semantic analysis," *Inf. Process. Manage. 41(1): 75-95*, 2005.

[8] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields," in *IJCAI: 2862-2867*, 2007.

[9] W. Gao, P. Li, and K. Darwish, "Joint topic modeling for event summarization across news and social media streams," in *CIKM:1173-1182*, 2012.

[10] Y. Lu, C. Zhai, and N. Sundaresan, "Rated aspect summarization of short comments," in *WWW: 131-140*, 2009.

[11] X. Wan and J. Yang, "Multi-document summarization using cluster-based link analysis," in *SIGIR: 299-306*, 2008.

[12] P. Hu, C. Sun, L. Wu, D.-H. Ji, and C. Teng, "Social summarization via automatically discovered social context," in *IJCNLP: 483-490*, 2011.

[13] M. Hu, A. Sun, and E.-P. Lim, "Comments-oriented document summarization: Understanding document with readers' feedback," in *SIGIR: 291-298*, 2008.

[14] M.-T. Nguyen, C.-X. Tran, D.-V. Tran, and M.-L. Nguyen, "Solscsum: A linked sentence-comment dataset for social context summarization," in *CIKM*, 2016.

[15] T. Joachims, "Training linear svms in linear time," in *KDD: 217-226*, 2006.

[16] M.-T. Nguyen, V.-A. Phan, T.-S. Nguyen, and M.-L. Nguyen:, "Learning to rank questions for community question answering with ranking svm," in *CoRR abs/1608.04185*, 2016.

[17] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research, 3: 993-1022*, 2003.

[18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS: 3111-3119*, 2013.

[19] M.-T. Nguyen, Q.-T. Ha, T.-D. Nguyen, T.-T. Nguyen, and L.-M. Nguyen, "Recognizing textual entailment in vietnamese text: An experimental study," in *KSE: 108-113*, 2015.

[20] M. F. Porter, "Snowball: A language for stemming algorithms," 2011.

[21] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning 20(3): 273-297*, 1995.

[22] A. Nenkova, "Automatic text summarization of newswire: lessons learned from the document understanding conference," in *AAAI: 1436-1441*, 2005.

[23] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research, 22: 457-479*, 2004.

[24] Y. Freund, R. D. Lyeryer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *Journal of Machine Learning Research 4: 933-969*, 2003.

[25] C.-Y. Lin and E. H. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *HLT-NAACL: 71-78*, 2003.

[26] M.-T. Nguyen, A. Kitamoto, and T.-T. Nguyen, "Tsum4act: A framework for retrieving and summarizing actionable tweets during a disaster for reaction," in *PAKDD (2): 64-75*, 2015.