International Journal on Artificial Intelligence Tools

Exploiting User-Generated Content to Enrich Web Document Summarization*

Minh-Tien Nguyen[†], Duc-Vu Tran, Chien-Xuan Tran, and Minh-Le Nguyen[‡]
Japan Advanced Institute of Science and Technology,
1-8 Asahidai, Nomi, Ishikawa, 923-1292, Japan.
{*tiennm, vu.tran, chien-tran, nguyenml*}*@jaist.ac.jp*

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

User-generated content such as comments or tweets (also called by social information) following a Web document provides additional information for enriching the content of an event mentioned in sentences. This paper presents a framework named *SoSVMRank*, which integrates the user-generated content 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 comments or tweets are exploited to support sentences in a mutual reinforcement fashion. To model sentence-comment (or tweet) relation, a set of local and social features are proposed. After ranking, top m ranked sentences and comments (or tweets) are selected as the summarization. To validate the efficiency of our framework, sentence and story highlight extraction tasks were taken as a case study on three datasets in two languages, English and Vietnamese. Experimental results indicate that: (i) our new features improves the summary performance of the framework in term of ROUGE-scores compared to state-of-the-art baselines and (ii) the integration of user-generated content benefits single-document summarization.

Keywords: Information Retrieval; Learning to Rank; Web Document Summarization, Social Context Summarization, NLP.

1. Introduction

The rapid growth of information on the Internet provides the large amount of online data (usually called by big data) in different formats, from many sources. One of the advantage of big data is that people can quickly follow new information of any event in any domain, e.g. travel, sport via the data spread. On the other hand, they are also overwhelmed by their daily exposure and such beneficial use is challenged by the characteristics of big data such as diversity, noise. These challenges

*This manuscript is an improved and extended version of the paper: Learning to Summarize Web Documents using Social information, presented at 28^{th} *International Conference on Tools with Application Intelligence* (ICTAI) 2016, San Jose, California, USA.
[†]Hung Yen University of Technology and Education (UTEHY), Vietnam.
[‡]Corresponding Author.

demand sophisticated text summarization systems, which process raw data to distill important information, e.g. summary sentences.

In the context of social media, Web 2.0 generation, e.g. Yahoo News[a] 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 discuss the event by writing their comments on the Web interface. At the same time, users also update the progress of the event by posting messages (tweets) on their timeline in a social network, e.g. Twitter. After posting these messages (we define as user-generated content), their friends or other readers or can immediately update the news content. Fig. 1 shows a generic scheme of news and social media. The user-generated content, one form of social

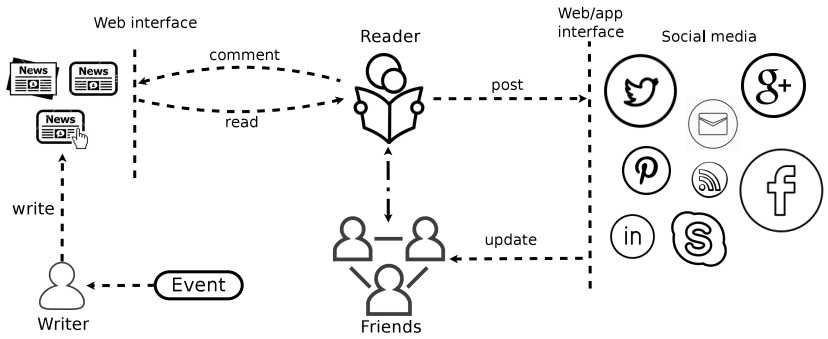


Fig. 1. The generic scheme of news and social media.

information,[1,3,4,5,28] has important characteristics. Firstly, the content written or posted by readers after reading a Web document is closely related to the story of an event. Secondly, sentences from several messages are well written and can be directly used as a summary unit. Finally, the messages with the opinions of readers can be utilized to enrich summary sentences in the original document. These characteristics inspire a novel summary task which exploits the social information of a Web document to support sentences for generating the summarization.

This study presents a framework which automatically extracts summary sentences and comments (or tweets) of a Web document by incorporating its social information. In order to do that, we present the extraction in the form of learning to rank (L2R), in which we train a L2R-based summary model and rank to select top m ranked sentences and comments as the summarization. In modeling, sentence-comment (or tweet) relation is denoted in a mutual reinforcement fashion. This view allows to consider sentences as the support when modeling a comment (or a tweet). This paper makes the following contributions:

- It presents the sentence-comment (or tweet) relation in a mutual reinforcement fashion. The presentation allows to integrate the support of user-

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

generated content into the summary process.
- It proposes sophisticated features to integrate social information into the summary process. The features are different from Refs.[4,1]
- It carefully conducts an investigation to show the impact of each feature and feature group. The investigation provides a better understanding in selecting appropriate features for social context summarization.
- It releases a dataset[b] which is derived from Ref.[4] The human involve to annotate this data to create labels for each sentence and tweet. The annotated data can be used to train supervised learning methods.
- It presents a unified framework[c] which utilizes sophisticated features in a mutual reinforcement fashion. Our architecture framework is straightforward to incorporate additional features.

In next sections, we first introduce related works and definitions used in our study. Next, we present our framework along with basic idea, data preparation, and observation. We also describe the framework in three steps: basic model, social context integration, and summarization. After generating the summarization, we show experimental results with discussion and deep analyses. We finish by drawing important conclusion.

2. Literature Review

2.1. *Text Summarization*

Text summarization has receipted lots of attention from scientists. It has been addressed by unsupervised learning methods[6,7] or supervised learning ones.[9,10,11] In the last decade, many researchers have applied machine learning to text summarization using classification with features,[8] the combination of classification and latent semantic analysis,[10] or sequence labeling with a set of features.[11] They have achieved promising results, e.g. 0.483 in ROUGE-2[d] (Recall-Oriented Understudy for Gisting Evaluation: 1-gram or 2-gram based co-occurrence statistics) and 0.419 in F-1 on DUC 2001 dataset.[11] Recently, the summarization has been presented in the form of submodular functions, in which each function guarantees two aspects: the representative and diversity.[12] Another direction is that the summarization has been also denoted in the form of concepts presented by phrases and is generated by using integer linear programming (ILP).[13] There are also many studies which focus on extracting summary sentences in a document such as LexRank,[14] or deep learning.[15,16,17]

2.2. *Social Context Summarization*

Social information such as hyper-links were first exploited by Refs.[18,19] for summarization. The summarization was generated by selecting sentences in hyper-links.

[b]Download at: http://150.65.242.101:9292/yahoo-news.zip
[c]http://150.65.242.101:9293/?paper=ijait
[d]https://en.wikipedia.org/wiki/ROUGE_(metric)

Click-through data[20] was also used as a help to extract summary sentences by significant words[6] and latent semantic analysis.[25] This method assumes that query keywords from users typed on search engines usually reflect the content of a Web document. These methods face three challenging issues: (i) only extracting summary from hyper-links,[18,19] (ii) there are no links from a new Web page to the older ones and (iii) pages which Web users click on may be irrelevant to their interest.[20] Later, user-generated content such as comments was also used to support sentences for generating the summarization. The extraction was done by using different approaches such as clustering with feature vectors[21], selecting sentences based on their distance to graphs which present topics discussed among readers[22], or rated aspect summarization for a target object.[23]

Recently, tweets from Twitter were widely used to support sentences in generating the summarization in supervised[5,4] or unsupervised learning fashion.[24,3,1,26,2] For example, in the supervised learning fashion, the extraction can use Support Vector Machines (SVM) and Conditional Random Fields (CRF) as preliminary steps, and then builds a dual wing factor graph model for extracting summaries.[5] Another study uses learning to rank (L2R) trained by RankBoost[32] with a set of features including local sentence, local tweet, and cross features for news highlight extract.[4] On the other hand, topic modeling can be utilized to extract summaries.[24] The authors proposed a cross-collection topic-aspect modeling (cc-TAM) exploited the cc-TAM as a preliminary step to generate a bipartite graph used by co-ranking to select sentences and tweets for multi-document summarization. Graph-based approach such as LexRank could also used to integrate tweets into the summary process.[3] The authors introduced a variation of LexRank, which uses auxiliary tweets for building a heterogeneous graph random walk (HGRW) to summarize single documents. Features-driven in the form of graph-based method was also used.[1,2] The authors present a framework, which combines inter-relation and intra-relation in a mutual reinforcement fashion to score each sentence and tweet. An ILP-based extraction was presented by exploring public posts following a new article to improve automatic summary generation.[26] The authors define different approaches to incorporate information from public posts from Facebook to estimate bi-gram weight in the form of ILP. Finally summary contains a set of bi-grams.

The previous methods exist two issues: (i) unsupervised methods, e.g HGRW are sensitive to data and (ii) several methods only select sentences or social messages as the summarization. We claim that the summarization should include summary sentences from both Web documents and their social information because social information can enrich the content of sentences in several aspects, e.g. the viewpoint from readers who involve the event. Our method addresses the two issues, in which, firstly, we propose a supervised framework which integrates the human knowledge into the summary process. Secondly, the summarization in our method contains both sentences and comment (or tweets) instead of only selecting sentences.

3. Definition

User-generated Content: In this study, user-generated content is defined as 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, the user-generated content of d is denoted by UG_d generated 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 created by U after reading d.

Social Content: We follow Ref.[5] to define the social context of a Web document d is C_d represented by $\langle S_d, UG_d, U_d \rangle$, where S_d is a set of sentences in document d, UG_d is a set of tweets or comments on d written by users U_d. In this work, user relation is eliminated because it is an implicit factor and is unavailable in datasets.

Social Content Summarization: Social context summarization is to select summary sentences and representative user-generated content such as comments or tweets as the summarization by using C_d giving the original document d.

4. Summarization with User-Generated Content

This section shows our proposal to select summary sentences and comments or tweets of a Web document by incorporating its social context. We first present our idea and summary framework. Next, we show data preparation and observation. Finally, we describe the proposed method for achieving the objective, and show evaluation metric used to compare our method to state-of-the-art baselines.

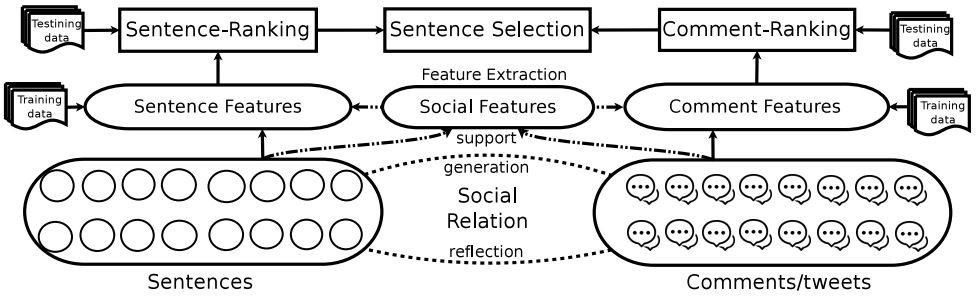


Fig. 2. The overview of SoSVMRank, in which a document contains a set of sentences and comments. A sentence is presented by a set of words denoted by circles; the dot lines connect sentences, comments to social features show the mutual support of sentences and comments.

4.1. *Basic Idea*

We formulate our problem in the form of learning to rank (L2R), in which the summarization is generated by ranking sentences based on their informative information. The 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 indicative information. Fig. 2 presents our framework.

When modeling a sentence, the framework exploits a set of social features from comments to support local features, e.g. the dot line from comments. Similarly, a set of social features from sentences is also used to enrich local features in modeling a comment. This view allows to consider sentences as social information of comments. In this model, local and social information support together in a mutual reinforcement fashion. After modeling, two L2R-based summary models are separately trained on training data and applied to testing data. Finally, the framework selects top m ranked sentences and comments as the summarization.

Our approach distinguishes the traditional methods[8,9,10,11,5] in two aspects: (i) we formulate the summarization as a L2R task, (i) we exploit the support of user-generated content to enrich information of sentences in a mutual reinforcement form in feature extraction. Our model shares the idea of using L2R with Refs.[4,27]; however, adding new features, analyzing feature contribution, and investigating the summarization by L2R methods are three key differences.

4.2. *Datasets*

DUC is well-known data for single or multiple document summarization; this dataset, however, lacks social information. We prepared three datasets in two languages named SoLSCSum,[30] USAToday-CNN,[4] and VSoLSCSum.[29]

SoLSCSum dataset: is an English dataset created for social context summarization.[30] The dataset contains 157 news articles along with 3,462 sentences, 5,858 gold-standard references and 25,633 comments. The label of each sentence or comment was created based on majority voting between annotators with the Cohen's Kappa[e] is 0.5845 with 95% confidence interval.

USAToday-CNN dataset: To confirm the efficiency of our framework, we used a news highlight extraction dataset derived from Ref.[4] The dataset contains 121 events along with 455 highlights and 78,419 tweets. The original dataset has no labels; therefore, to train supervised learning methods, e.g. SVM or CRF, we created label for each sentence and tweet. After removing near-duplicate tweets using Simpson formula, two annotators were asked via a web page[f] to give weak labels for each sentence or tweet. In this case, sentences and tweets with the labels are not gold-standard references (we use highlights for evaluation). To make the annotation easy, we provide a suggestion based on Cosine similarity. Each sentence or tweet is assigned a Cosine score (using bag-of-words model), which is the maximal score of a sentence or tweet with the highlights of a document. Label decision bases on both Cosine similarity suggestion and the content of a sentence or tweet. The selected sentences or tweets are no less than five and no more than 15 in total. Cohen's Kappa between annotators after validating is 0.617 with 95% confidence interval.

[e]http://graphpad.com/quickcalcs/kappa1.cfm
[f]http://150.65.242.91:9080/doc-sum-annotator/

VSoLSCSum dataset: To validate the performance of our model in non-English language, we used a Vietnamese dataset created for social context summarization.[29] The dataset consists of 141 open-domain articles along with 3,760 sentences, 2,448 gold-standard references, and 6,926 comments in 12 events. The agreement computed by Cohen's Kappa between the two annotators is 0.685 with 95% confidence interval. The strength of agreement is considered to be good.

4.3. *Data Observation*

We examined word overlapping between sentences and user-generated content. We considered each token as a single word and segmented all sentences and comments (for SoLSCSum and VSoLSCSum) and sentences and tweets (for USAToday-CNN), then counted the word overlapping of sentences over comments or tweets and vice versa. We did not observe word overlapping with stopword removal on VSoLSCSum due to no formal stopword list in Vietnamese. Table 1 shows the observation.

Table 1. Statistical observation on the two datasets; s: sentences, c: comments, and t: tweets.

Dataset	Observation	Sentences	Comments Tweets
SoLSCSum	% Token overlapping	s/c: 13.26	c/s: 42.05
	% Token overlapping with stopword removal	s/c: 8.90	c/s: 31.21
USAToday-CNN	% Token overlapping	s/t: 22.24	t/s: 16.94
	% Token overlapping with stopword removal	s/t: 15.61	t/s: 12.62
VSoLSCSum	% Token overlapping	s/c: 37.712	c/s: 44.820

From Tables 1, we observe that: (i) there exist common words or phrases between sentences and tweets or comments (called social messages) and (ii) readers tend to use words or phrases appearing in sentences to create their messages, e.g. 31.21% of word overlapping. From the observation, we consider four hypotheses:

- *Representation:* summary sentences in a Web document contain important information;
- *Reflection:* representative tweets or comments written by readers reflect document content as well as summary sentences;
- *Generation:* readers tend to use words or phrases appearing in a document to create their social messages, e.g. tweets or comments;
- *Common topic:* sentences and social messages mention common topics represented in the form of common words.

4.4. *Data Preparation*

We conducted a pre-processing step to remove comments and tweets with fewer than five tokens since they are fairly short for summarization. In SoLSCSum, 10-fold cross-validation was used with $m = 6$ (six summary sentences and six summary comments), the same setting with Ref.[30] 5-fold cross-validation was used with

$m = 4$ for USAToday-CNN, as a suggestion in Ref.[4,1] and 5-fold cross-validation with $m = 6$ for VSoLSCSum .[29] All summaries and gold-standard highlights were stemmed[g] using the stemming method.[31] Note that summaries in VSoLSCSum were not stemmed because there is no stemming method in Vietnamese.

4.5. *Summarization with Ranking SVM*

As mentioned, the objective of this study is to build a dual L2R-based summary model to extract summary sentences and comments (or tweets). This section describes our process to generate the summarization by using the support from social context. The process can be described in three steps: basic model with basic features, our model with new features, and summarization.

4.5.1. *Basic Model*

We started with a basic model shown in Ref.[4] In this study, the authors present a summary model, which integrates the support of tweets to enrich the summarization with a set of local and cross features. For example, when modeling a sentence, the cross features are exploited to support the local ones. The local features cover several aspects of a single sentence or tweet, e.g. sentence position, the importance score based on unigram hybrid term frequency – inverse document frequency (TF-IDF). The cross features exploit the mutual support from tweets when modeling a sentence, e.g. maximal Cosine score between a sentence. The detail of features is shown in Ref.[4] In our study, we ignore unnecessary features, e.g. URL or hashtags because (i) they are unavailable in comments (SoLSCSum and VSoLSCSum datasets) and (ii) they are inefficient due to the pre-processing step (USAToday-CNN dataset). To train the summary model, the authors use RankBoost.[32]

We extended the basic model in two aspects: (i) proposing new features, which can capture more characteristics of a sentence and (ii) using different learning algorithm. In the first aspect, it is understandable that adding more sophisticated features improve the summary performance. For example, the features in Ref.[4] do not consider sequence aspect, which plays an important role in summarization.[11] In the second aspect, because the basic model uses RankBoost, which bases on AdaBoost,[33] then the over-fitting problem may appear easily when training the summary model compared to other method, e.g. SVM.[34]

4.5.2. *Ranking SVM with New Features*

To train the L2R model, we adopted Ranking SVM,[35] one state-of-the-art L2R methods for information retrieval.[36] Ranking SVM[h] applies the characteristics of SVM[34] 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 $y_{(u,v)}^{(i)}$,

[g]http://snowball.tartarus.org/algorithms/porter/stemmer.html
[h]https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

Ranking SVM optimizes an objective function in Eq. (1):

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

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

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

where: $f(x) = w^T x$ is a linear scoring function, (x_u, x_v) is a pairwise and $\xi_{u,v}^{(i)}$ is the loss. The document pair-wise is sentence-sentence or comment-comment (tweet-tweet). In the SoLSCSum and VSoLSCSum datasets, the pair-wise order is the label of each sentence or comment. In the USAToday-CNN dataset, we follow (Ref.[4]) to define the pair-wise order as a salient score between a sentence or comment with the ground-truth summary sentences. For example, given $H = \{h_1, h_2, ..., h_k\}$ is a set of highlights generated by the human, Eq. (4) computes the salient score of a sentence or tweet.

$$score(s_i) = max\{\text{ROUGE-1 F-score}(s_i, h_j)\},\ j \in \{1, k\} \tag{4}$$

where: *ROUGE-1 F-score()* returns ROUGE-1 F-score between s_i and h_j.

New Local Features: Our first effort is to cover the *representation* hypothesis, which the basic features[4] may not completely consider. We present new features, that capture the inherent characteristics of a sentence and comment and cover four important aspects such as length, sequence, topic covering and meaningless words.

Sentence length: This feature bases on a hypothesis that a summary sentence usually contain more important information compared to non-summary ones. This feature counts the number of words in s_i.[i]

Sentence length before: The next four features cover sequence aspect in document. When writing, writers arrange sentences in an appropriate order to create a story. A summary sentence is followed by several supporting sentences, which enrich its meaning. Given a sentence s_i, we consider the previous and the next sentence because further sentences are the reference or do not directly support s_i. We first present the length of a previous sentence.

The sentence length before is the number of words in sentence s_{i-1} giving s_i. The value of this feature 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 of this feature is 0 if s_i is the last sentence in a document.

Cosine similarity before: shares the sequence aspect with the sentence length before and after features, in which we use Cosine similarity instead of the length aspect. Given sentence s_{i-1} and s_i denoted by vector $\overrightarrow{x}$ and $\overrightarrow{y}$ using the bag-of-

[i]When extracting features, all stopwords were removed.

words model, Eq. (5) defines the Cosine similarity of two vectors:

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

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 the two same size vectors. The cosine value of s_i is 0 if it is the first sentence in a document.

Cosine similarity after: calculates the similarity of s_i and s_{i+1} using Eq. (5). The value is 0 if s_i is the last sentence of a document.

By using the sentence length and Cosine similarity before and after, we consider the sequence aspect of the summarization, in which a summary sentence is selected based on its content represented by its surrounding sentences.

Local LDA score: This feature bases on the *common topic* hypothesis. It states that a summary sentence should include topics presented by topical words. It is possible to use TF-IDF to select important words; however, these words do not take into account topical aspect denoted in the form of word distribution. We, therefore, adopted Latent Dirichlet Allocation (LDA[j])[37] to generate the topical words. The generation was done in two steps: training and inference.

- *Training:* We first trained an LDA model[k] on a large number of news articles. In this view, we considered the LDA model as a global source. We used more than 280.000 articles[l] from DailyMail for training. Topic and word distribution parameter are $\alpha = \beta = 0.01$ with 1000 iterations; the topic number was empirically chosen $k = 150$. More precisely, we divided each document into two parts: d_s (sentences) and d_c (comments or tweets), then we formed each part as a smaller single document, that is an input for inference. After training, we obtained a word-topic-weight matrix and document-topic distribution matrix.
- *Inference:* Given a small single document d_s (or d_c), we first obtained top t closet topics of d_s. The closet topics are those that have the highest values in the document-topic distribution matrix. With each topic, we selected top w words, which have the highest weights in the word-topic-weight matrix. As a result, the number of topical words for each small single document is $(|t| \times |w|)$. In practice, we set $|t| = 5$ and $|w| = 5$, then the total number of topical words is 25. Given a set of topical words named $TF = \{w_1, ..., w_k\}$ inferred from d_s, Eq. (6) computes the local LDA score:

$$\textit{local-lda-score}(s_i) = \frac{\sum_{j=1}^{k} \textit{weight}(w_j)}{n} \text{ if } w_j \in s_i \tag{6}$$

  where: $weight()$ returns the word weight of w_j, e.g. 0.45 in s_i (normalized in [0, 1]) ; n is the number of words in s_i.

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

[k]compared to the original article, we train the LDA model on a larger corpus instead of only using datasets.

[l]http://homepages.inf.ed.ac.uk/s1537177/resources.html

Function words: The hypothesis of this feature is that a summary sentence should contain content words, which include important information, e.g. person name. Let len_{org} to be the length of an original sentence and len_{rmw} is the length of the sentence after removing stop words, Eq. (7) counts stop word number.

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

New Social Features: The local features only capture the internal aspect of a summary sentence, but they ignore the support from user-generated content, which provides additional information from readers. To improve the quality of our model, we present social features, which cover three aspects: semantic similarity, topical covering and entailment between a sentence and auxiliary comments or tweets.

Semantic-based similarity: This feature bases on the *generation* hypothesis, in which readers tend to use salient words in sentences to create their comments or tweets in a variation form. Table 2 shows the generation example taken from the event collected and posted by Yahoo News.[m]

Table 2. An example of the generation behaviour. The first sentence is in the Web document and the second one is a comment generated from readers.

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.
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.

In this example, we can observe that: (i) the sentence can be seen as a summary because it contains essential information; (ii) the comment also reflects the event and includes the viewpoint of readers; and (iii) the comment and sentence share several common words (bold words), which are directly extracted from the sentence, e.g. *"knife"*, *"officers"*, or are derived in a variation, e.g. *"fired"*, *"shots"*. From these observation, we present the sentence-comment relation by semantic similarity.

To exploit the semantic aspect, we present the relation by using *Word2Vec*.[38] The Word2Vec[n] takes a large dataset as an input and produces word vectors as the output. In training, the Word2Vec first generates a vocabulary from the dataset and maps each word in the vocabulary into a high-dimensional vector space, in which each vector represents the meaning of a word with its context. The context of a word is the number of its surrounding words (usually called by window size). After training, we can calculate the distance between two words, e.g. Cosine similarity between two words: *"police"* and *"officer"*. In practice, we trained a Word2Vec model on 1 billion words from Google by using SkipGram model, the vector dimension = 300 with the window size (word context) = 7.

[m]https://www.yahoo.com/news/boston-man-shot-police-target-terrorism-probe-officials-022407784.html?ref=gs
[n]https://code.google.com/p/word2vec/

Given the Word2Vec model, Eq. (8) calculates the semantic similarity of a sentence and auxiliary comments (or tweets):

$$w2v\text{-}score(s_i) = \max_{j=1}^{m} (sentSim(s_i, c_j)) \tag{8}$$

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

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

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 the *Word2Vec* model.

Social LDA score: This feature shares the characteristic of *Local LDA Score*, in which a summary should also cover topics discussed among readers. Given a set of topical words named $TF = \{w_1, ..., w_k\}$ inferred from d_c (the topical words were inferred from comments or tweets), the LDA score from social information is computed by Eq. (10) (the same mechanism as Eq. (6)).

$$aux\text{-}lda\text{-}score(s_i) = \frac{\sum_{j=1}^{k} weight(w_j)}{n} \text{ if } w_j \in s_i \tag{10}$$

where: n is the number of words in s_i, the word weight is derived from topical words on the social side. By combining the two LDA scores, our model states that a summary sentence should not only cover topics written by writers on the document side but also include topics discussed among readers on the social side.

Distance-based similarity: This feature tackles the *generation* hypothesis by formulating the sentence-comment relation in the form of recognizing textual entailment (RTE).[39] The RTE is a task which decides whether the meaning of a text can be plausibly inferred from another text in the same context.[39] This feature treats a different aspect compared to the *semantic-based similarity*, in which it operates on word and lexical level instead of semantic level. We present a distance-based feature, which bases on a set of distance features derived from Ref.[1,2] Table 3 shows the features. The detail of each feature can be seen in Ref.[2]

Table 3. The features; S: a sentence, C: a comment or tweet.

Distance Features	**Lexical Features**
Manhattan distance	The longest common sub string of S and C
Euclidean distance	Inclusion-exclusion coefficient
Cosine similarity	% words of S in C
Word matching	% words of C in S
Dice coefficient	Word overlap coefficient
Jaccard coefficient	—
Jaro coefficient	—
Damerau-Levenshtein	—
Levenshtein distance	—

The distance feature states that a summary sentence and comment should be closer compared to non-summary ones. To compute the distance between two vectors, a similarity score, e.g. Cosine can be used; however, using a single measurement may not efficient enough to completely capture the similarity aspect of a sentence-comment pair. For example, a summary sentence and comment may not share common words due to content variation, which may negatively affect the Cosine calculation. We, therefore, consider the word and character level of a sentence-comment pair by using various distance features. For example, the Manhattan distance covers pairs those share common words and the Levenshtein distance based on characters treats pairs; those are content variation in sharing common characters. Eq. (11) presents the distance of a sentence and auxiliary comments:

$$dist(s_i) = \max_{j=1}^{m} (distSim(s_i, c_j)) \tag{11}$$

where: m is the number of comments; $distSim()$ returns the distance similarity of s_i and c_j and is computed by Eq. (12):

$$distSim(s_i, c_j) = \frac{1}{F} \sum_{n=1}^{F} f_n(s_i, c_j) \tag{12}$$

where: F contains nine distance features in Table 3; $f_n()$ is a similarity function computed by each n^{th} feature.

Lexical-based similarity: This feature also shares the *generation* hypothesis with *distance-based similarity* but using common word aspect. It states that a summary sentence and comment should share common words. This feature was modeled in the same mechanism with the distance feature but using five lexical features in Table 3. Note that our proposed features were also used for modeling comments.

4.5.3. *Summarization*

After training, our model was applied to the testing set to select the summarization. Equation (13) presents the selection.

$$S_r \leftarrow ranking(S); \quad C_r \leftarrow ranking(C) \tag{13}$$

where: *ranking()* returns a list of sentences or comments (tweets) in a decreased weight order. After ranking, top m ranked sentences and comments from S_r and C_r are selected as the summarization.

4.6. *Statistical Analysis*

This section introduces baselines used to compare to our method and evaluation metric used to compare our method against the baselines.

4.6.1. *Baseline*

We compared SoSVMRank to state-of-the-art methods in social context summarization. These methods are listed as the following:

- **SentenceLead-m:** chooses the first m sentences as the summarization.[40] This method was not used in selecting comments or tweets.
- **LexRank**[14]**:** was proposed for document summarization. This method builds a stochastic graph-based method for computing relative importance of textual units in text summarization. In this study, LexRank algorithm[o] was applied with tokenization and stemming.[p]
- **Cosine-based ILP**[13]**:** was used for sentence extraction. We adopted this method by using the average Cosine score as the weight of a sentence with top 10 important words. The important words are those which have the highest TF-IDF scores.[q] This method was separately used for selecting summary sentences and comments or tweets.
- **SVM**[34]**:** was used for text summarization.[5] We train a binary classifier by using LibSVM[r] with RBF kernel. The features were derived from Ref.[5] and scaled in [-1, 1]; comments were weighted by 85%.[30]
- **CRF**[41]**:** was used for single document summarization.[11] This method formulates the summarization as a sequence labeling task, in which summary sentences are labeled by 1 and non-summary sentences are labeled by 0. Features were derived from Ref.[11]
- **SoRTESum**[1]**:** combines intra-relation and inter-relation to score each sentence and comment (or tweet) in a mutual reinforcement fashion. This method includes two models: using inter information (SoRTESum Inter Wing) and the both relations (SoRTESum Dual Wing).
- **Ranking SVM**[35]: We separately trained two L2R models ($C = 3$ with linear kernel) for sentences and comments (or tweets) by using the features from Ref.[4] without the use of new features in Section 4.5.2.

4.6.2. *Evaluation Metric*

In SoLSCSum and VSoLSCSum, we used selected sentences and comments (those which were labeled by 1 in the annotation step) as gold-standard references. In USAToday-CNN dataset, highlights were used as gold-standard references. For evaluation, F-1 ROUGE-N[42](N=1, 2)[s] was employed as the defination in Eq. (14):

[o]https://code.google.com/p/louie-nlp/source/browse/trunk/louie-ml/src/main/java/org/louie/ml/lexrank/?r=10
[p]http://nlp.stanford.edu/software/corenlp.shtml
[q]when calculating the TF-IDF, we consider each sentence as a single document and all sentences in a document as a set of documents
[r]http://www.csie.ntu.edu.tw/∼cjlin/libsvm/
[s]http://kavita-ganesan.com/content/rouge-2.0-documentation

$$ROUGE - N = \frac{\sum_{s \in S_{ref}} \sum_{gram_n \in s} Count_{match}(gram_n)}{\sum_{s \in S_{ref}} \sum_{gram_n \in s} Count(gram_n)} \quad (14)$$

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 reference summaries, $Count(gram_n)$ is the number of n-grams in the reference summaries.

5. Results and Discussion

This section firsts show comparison results of SoSVMRank with state-of-the-art baselines. Section 5.2 investigates the feature contribution in our model. We also present the sentence length and position distribution observation. We finally validate our hypotheses and deeply analyze our model by a running example.

5.1. *Experimental Results*

We report results on Web document summarization using social context in Table 4 with ROUGE-1 and ROUGE-2 F-score on the three datasets. Results in Table 4 show that our method obtains improvements of ROUGE-scores compared to the baselines. For example, our method is the best in SoLSCSum and VSoLSCSum except for ROUGE-2 of sentence extraction in SoLSCSum. In USAToday-CNN, our model also comparably performs the baselines. This is because highlights in USAToday-CNN were generated by human in an abstract fashion, which challenges our features. However, in ROUGE-1 of tweet extraction, our method is the best, e.g. 0.233 vs. 0.226. These results confirm the efficiency of our method and features, and support the hypotheses stated in Section 4.3.

Our method slightly outperforms the basic model (SVMRank with basic features) in almost cases, except for ROUGE-1 and ROUGE-2 of sentence selection in USAToday-CNN dataset. It is because features in Ref.[4] also capture well the summary aspect in each sentence. However, compared to the original model,[4] which uses RankBoost named L2R CCF (see Table 5), our method obtains significant improvements, e.g. 0.425 vs. 0.363. CRF comparably performs other methods in sentence selection, but it achieves very poor results in comment or tweet extraction, e.g. 0.070 vs. 0.367 in ROUGE-1 of comment extraction in VSoLSCSum. This is because the sequence aspect may not explicitly exist in social messages; therefore, it limits CRF. SoRTESum is a competitive method over the three datasets even it is an unsupervised learning method. This suggests that exploiting social information in an appropriate fashion benefits the summarization. Sentence Lead[40] is a strong baseline because it formulates the summarization by taking some first sentences.

We also compared our model to state-of-the-art methods in social context summarization on the three datasets. The trend in Table 5 is similar to Table 4, in which our method is the best in almost cases. For example, our method significantly outperforms HGRW[24] in ROUGE-1 of sentence extraction on SoLSCSum,

Table 4. Summary performance; *: supervised method; **bold**: the best value; *italic*: the second best; Lead-m was not used for comments or tweets. Methods with S use social information.

Dataset	Method	Document		Comment Tweet	
		ROUGE-1	**ROUGE-2**	**ROUGE-1**	**ROUGE-2**
SoLSCSum	SentenceLead-m	0.365	0.322	—	—
	LexRank	0.328	0.257	0.244	0.140
	Cosine-based ILP	0.195	0.108	0.097	0.033
	SVM*	0.293	0.239	0.141	0.074
	CRF*	0.413	**0.391**	0.074	0.062
	SoRTESum IW (S)	0.357	0.299	0.237	0.135
	SoRTESum DW (S)	0.362	0.302	0.206	0.113
	SVMRank* (S)	*0.420*	0.317	*0.365*	*0.154*
	Our method* (S)	**0.425**	*0.323*	**0.371**	**0.158**
USAToday-CNN	SentenceLead-m	0.249	*0.096*	—	—
	LexRank	0.183	0.045	0.154	0.056
	Cosine-based ILP	0.229	0.054	0.198	0.052
	SVM*	*0.262*	0.088	0.216	*0.073*
	CRF*	0.232	0.062	0.189	0.052
	SoRTESum IW (S)	0.255	**0.098**	0.201	0.068
	SoRTESum DW (S)	0.254	*0.096*	0.209	**0.074**
	SVMRank* (S)	**0.279**	0.079	*0.226*	0.063
	Our method* (S)	*0.261*	0.062	**0.233**	0.071
VSoLSCSum	SentenceLead-m	0.437	0.393	—	—
	LexRank	0.471	0.381	0.344	0.246
	Cosine-based ILP	0.283	0.221	0.128	0.076
	SVM*	0.505	0.438	0.324	0.181
	CRF*	0.378	0.341	0.070	0.052
	SoRTESum IW (S)	0.471	0.383	0.336	0.233
	SoRTESum DW (S)	0.486	0.427	0.296	0.203
	SVMRank* (S)	*0.525*	*0.478*	*0.364*	*0.266*
	Our method* (S)	**0.534**	**0.489**	**0.367**	**0.268**

Table 5. Our method vs. state-of-the-art methods on the three datasets.

Dataset	Method	Document		Comment Tweet	
		ROUGE-1	**ROUGE-2**	**ROUGE-1**	**ROUGE-2**
SoLSCSum	cc-TAM[24]	0.321	0.268	0.166	0.088
	HGRW[3]	*0.377*	*0.321*	*0.248*	*0.145*
	L2R* (CCF)[4]	0.363	*0.321*	0.217	0.111
	Our method*	**0.425**	**0.323**	**0.371**	**0.158**
USAToday-CNN	cc-TAM[24]	0.261	*0.074*	**0.248**	*0.071*
	HGRW[3]	**0.271**	**0.091**	0.207	0.053
	L2R* (CCF)[4]	0.251	0.069	*0.238*	**0.076**
	Our method*	*0.261*	0.062	0.233	*0.071*
VSoLSCSum	cc-TAM[24]	0.405	0.336	0.199	0.125
	HGRW[3]	*0.514*	0.438	*0.362*	*0.265*
	L2R* (CCF)[4]	0.507	*0.447*	0.355	0.259
	Our method*	**0.534**	**0.489**	**0.367**	**0.268**

i.e. 0.425 vs. 0.377. This is because: (i) our model integrates the human knowledge in the form of features and (ii) our method is a supervised learning method instead of ranking based on random walk graphs. This shows that HGRW is a

competitive method even it is unsupervised. For example, it is the second best on SoLSCSum and VSoLSCSum and the best in sentence selection on USAToday-CNN. L2R (CCF) based on RankBoost also comparably performs other methods showing the efficiency of features in Ref.[4] cc-TAM[24] achieves quite poor results because it is designed for multi-document summarization whereas the three dataset are for single document summarization. The ROUGE-scores in both Tables 4 and 5 indicate that our method may be limited on USAToday-CNN. As mentioned, highlights in this dataset were generated by the human in an abstract way; therefore, more sophisticated features should be considered to tackle the abstract aspect.

5.2. *Feature Contribution*

We observed the contribution of our new features[t] by averaging feature weight generated from the model in each fold on SoLSCSum dataset.

Table 6. Our feature contribution in document summarization.

Document summarization			
Local Features	**Feature Weight**	**Social Features**	**Feature Weight**
Sent-length	1.533	Semantic-based score	1.768
Sent-length before	-0.283	Aux-LDA score	-0.101
Sent-length after	-0.251	Lexical-based sim	0.024
Cosine similarity before	1.039	Distance-based sim	0.216
Cosine similarity after	1.044	—	—
Local LDA score	-0.267	—	—
Function word	0.489	—	—

Table 7. Our feature contribution in comment summarization.

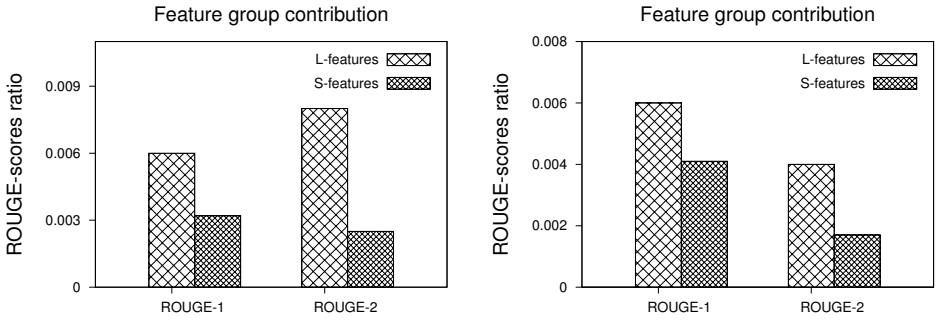
Comment summarization			
Local Features	**Feature Weight**	**Social Features**	**Feature Weight**
Sent-length	-0.213	Semantic-based score	0.085
Sent-length before	-0.655	Aux-LDA score	-0.153
Sent-length after	-0.317	Lexical-based sim	2.197
Cosine similarity before	0.475	Distance-based sim	0.270
Cosine similarity after	0.217	—	—
Local LDA score	-0.415	—	—
Function word	-1.155	—	—

Feature weight in Tables 6 and 7 indicates 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 entailment score also play an important role in our model whereas auxiliary topical score is negative. This trend is similar to comment extraction. Interestingly, the sentence length and function word are positive in sentence selection but they are negative in comment extraction. It is understandable that long comments usually include redundant information, e.g. the

[t]The contribution of basic features can be seen in Ref.[4]

opinion of readers. For the function word, because comments or tweets are written in an informal style with noise, then counting the number of stop words is inefficient.

We further investigated the contribution of each feature group in our model. We combined our new features with the basic features and run the model with three settings: (i) using all features, (ii) using local features (new and old features), and (iii) using social features (new and old features). The influence of each group is defined as the ratio of ROUGE-score F-1 computed by the ROUGE-score F-1 minus of the first setting for the second and the third setting.



(a) Sentence selection (b) Comment extraction

Fig. 3. The contribution of feature groups

Figure 3 shows that both local and social features contribute the summarization with positive values. It means that when removing these features, the summary performance is decreased. The values of the local features are larger than that of the social features indicate that the inherent information of each sentence or comment is more important than social information. It is understandable that the model uses many sophisticated features from a sentence as a main part and exploits additional features from comments or tweets as the support. The social features slightly affect the framework in sentence selection with tiny values; however, in comment extraction the contribution of social features increases, but it is not significant. This change explains the results in Table 4, where SoSVMRank slightly outperforms the basic model, e.g. 0.425 vs. 0.420 in ROUGE-1 of sentence selection on SoLSCSum or 0.233 vs. 0.226 in ROUGE-1 of tweet extraction on USAToday-CNN dataset.

We also confirmed the efficient of our features in different L2R methods on SoLSCSum dataset. We run RankBoost[32] by using defalt setting (iteration = 300, metric is ERR10) implemented in RankLib[u] and Ranking SVM[35] ($C = 3$ with linear kernel) with two settings: (i) using basic features and (ii) using all features with the same data segmentation. ROUGE-scores in Table 8 indicate that our features improve the summary performance on both two L2R methods. Ranking SVM with all features slightly outperforms the basic model, e.g. 0.425 vs. 0.420, but the new

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

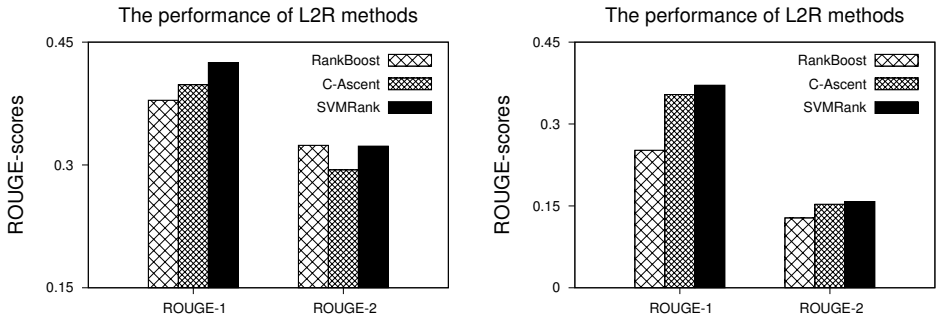
Table 8. Our feature contribution on two L2R methods.

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

features boost the ROUGE-scores of RankBoost, e.g. 0.252 vs. 0.217 in ROUGE-1 of comment extraction. The general trend of ROUGE-scores on two L2R methods confirms the efficiency of our features.

5.3. *Summary Performance with L2R Methods*

We investigated the affect of L2R methods to our features by running three L2R methods: RankBoost[32] (iteration = 300, metric is ERR10), Coordinate Ascent (random restart = 2, iteration = 25, tolerance = 0.001 with non-regularization), SVMRank[35] ($C = 3$ with linear kernel). We used all features (the basic and our new features) to train summary models on SoLSCSum.



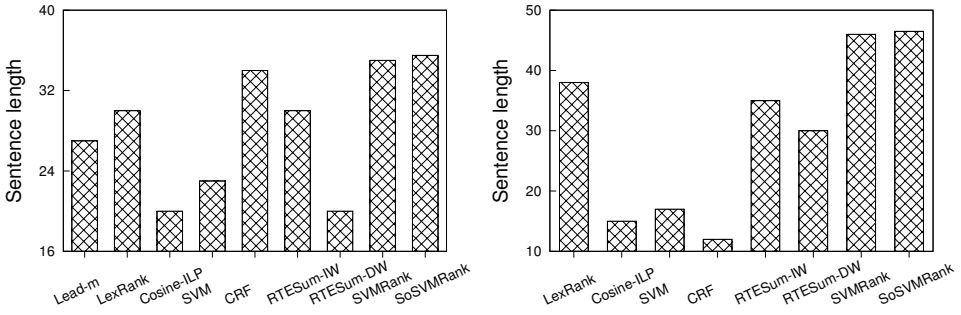
(a) Sentence selection (b) Comment extraction

Fig. 4. ROUGE-scores of L2R methods.

The ROUGE-scores in Fig. 4 indicate that Ranking SVM is the best in both sentence and comment extraction. It is understandable that Ranking SVM inherits powerful characteristics from SVM to perform pair-wise ranking. 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 over-fitting problem, which can appear in other methods, e.g. RankBoost. Coordinate Ascent comparably performs Ranking SVM except for ROUGE-2 of sentence selection. RankBoost achieves the worst performance in almost cases even it uses all the features. This supports our idea stated in Section 4.1, in which we not only improve the basic model by adding sophisticated features but also exploiting a strong L2R method.

5.4. *Sentence Length Observation*

We observed the length of output summaries from all the methods to show the relation between sentence length and summary performance on SoLSCSum dataset. The length was computed by the average of extracted sentences or comments of each method in each fold.



(a) The average of sentence length (b) The average of comment length

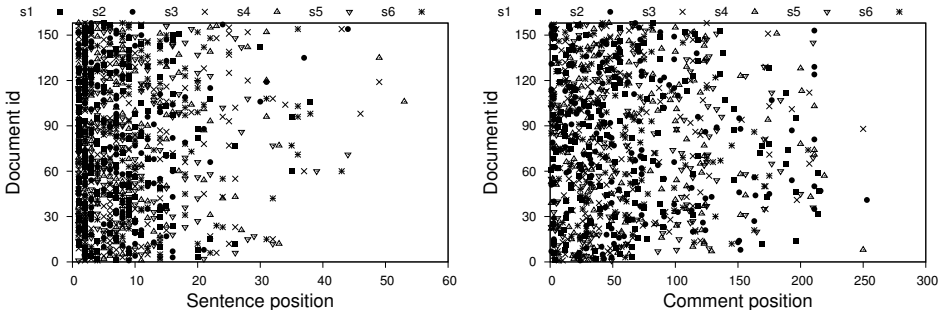
Fig. 5. The average of sentence and comment length

Figures 5(a) and 5(b) indicate that longer sentences belong to the best models, i.e. CRF, SoRTESum, or SVMRank. SoSVMRank generates the longest sentences, i.e. 35.5 in document summarization and 46.5 on comment extraction. This supports the results in Table 4 because longer sentences or comments usually contain more information compared to the shorter ones. In document summarization, CRF and SVMRank generate sentence length as similar as our method, but in comment extraction, CRF outputs shortest comments because sequence aspect dost not explicitly exist in comments. Other models obtain a similar sentence length, e.g. LexRank, SoRTESum and L2R while poor models achieve shorter length, i.e. SVM or Cosine ILP. This trend is similar to Figure 5(b).

5.5. *Sentence Position Observation*

We further observed the positions of extracted sentences and comments generated from SoSVMRank on SoLSCSum dataset. We collected summaries and matched them again to the original documents to reveal their positions.

From Figures 6(a) and 6(b), 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 6(a) and 252^{th} in Figure 6(b) because some documents contains a larger number of sentences and comments. Considering the data observation in Table 1, we conclude that the density distribution of comments is scattered because sequence aspect does not explicitly exist on the comment side. This explains the reason that CRF obtains poor results on comments and tweets.



(a) Extracted sentence position (b) Extracted comment position

Fig. 6. Sentence and comment position distribution

5.6. *Hypothesis Analysis*

We validated our hypotheses stated in Section 4.3 by running an example generated from SoRTESum Dual Wing[1] method on USAToday-CNN dataset. Table 9 shows the example. We simulated the running process of this method in Fig. 7, in which a sentence is denoted by a set of words represented in ovals.

Table 9. Two sentences and tweets taken from USAToday-CNN dataset; S_1 and T_2 are summary sentences, and S_2 and T_1 are non-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.

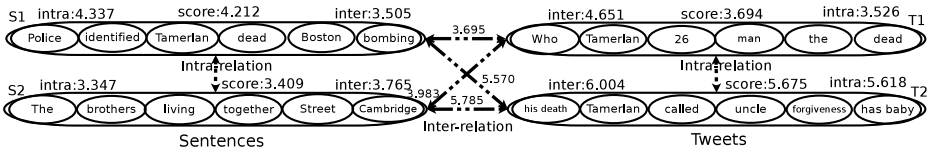


Fig. 7. A running example from Table 9 implemented SoRTESum Dual Wing.

In Table 9, S_1 supports the *representation* hypothesis, in which it contains important information of the Boston bombing event. People can partly understand the event by reading S_1. T_1 and T_2 confirm the *reflection* hypothesis, in which they also include salient information of the event. In addition, T_2 provides additional information (he has a baby), which is common in comments or tweets because readers usually express their viewpoints of an event. The bold words in sentences and tweets validate the *generation* and *common topic* hypothesis, in which readers tend to borrow salient words from sentences to create their messages. The simulation in

Fig. 7 also reveals that sentences with salient words obtain high scores compared to non-summary ones, e.g. S_1 and T_2 receive higher scores compared to S_2 and T_1.

5.7. *Error Analysis*

A further analysis of extracted sentences generated from our model was intensively conducted on SoLSCSum dataset. In Table 10, 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 10 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.

6. Conclusion

This paper introduces SoSVMRank to address social context summarization. The framework models the summarization as a L2R task and exploits the support from user-generated content to enrich sentences in a mutual reinforcement fashion. This paper also presents new features used to modeling the sentence-comment relation. After ranking, top m ranked sentences and user-generated content are selected as the summarization. Applying the framework to two tasks: sentence extraction and story highlight generation of single document suggests that it can be viable alternative to extraction-based systems. Promising results indicate that: (i) formulating sentence extraction in the form of L2R benefits the summarization and (ii) our features are efficient for single-document summarization.

For future directions, an obvious step is to investigate how the framework works to other domains and text genres. Our model is straightforward to integrate any additional features such as rhetorical relations between sentences or tree edit distance between sentences and user-generated content.

Table 10. Extracted summaries of document 121^{th} on SoLSCSum dataset.

Summary	
Sentences	**Comments**
[+]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."	[+]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.
[-]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".	[+]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.
[+]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.	[+]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.
[-]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.	[-]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.
[+]S5: A man who identified himself on Twitter as Rahim's brother said the family was shocked by the shooting.	[+]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.
[+]S6: "The FBI and the Boston Police did everything they could to get this individual to drop his knife," Evans said.	[+]C6: Yet his Iman brother was already claiming he was shot in the back with this hands in the air.

Acknowledgments

We would like to thank Gao and Li for sharing the code.[24] This work was supported by JSPS KAKENHI Grant numbers 15K16048 and JP15K12094, and CREST, JST.

References

1. M. T. Nguyen and M. L. Nguyen, Sortesum: A social context framework for single-document summarization, in *European Conference on Information Retrieval (ECIR'16)*, (2016), pp. 3-14.
2. M. T. Nguyen and M. L. Nguyen, Intra-relation or inter-relation?: Exploiting social information for Web document summarization, *Expert Systems with Applications* **76** (2017) 71-84.
3. Z. Wei and W. Gao, 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'15)*, (2015), pp. 1003-1006.
4. Z. Wei and W. Gao, Utilizing microblogs for automatic news highlights extraction, in *COLING* (2014), pp. 872-883.
5. Z. Yang, K. Cai, J. Tang, L. Zhang, Z. Su and J. Li, Social context summarization, in *Proceedings of the 34th international ACM SIGIR conference on Research and*

development in Information Retrieval (SIGIR'11) (2011), pp. 255-264.
6. H. P. Luhn, The automatic creation of literature abstracts *IBM Journal of Research Development* **2**(2) (1958) 159-165.
7. H. P. Edmundson, New methods in automatic extracting, *Journal of the Association for Computing Machinery* **16**(2) (1969) 264-285.
8. R. J. Kupiec, J. O. Pedersen and F. Chen, A trainable document summarizer, in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'95)* (1995), pp. 68-73.
9. M. Osborne, Using maximum entropy for sentence extraction, in *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4* (2002), pp. 1-8.
10. J. Y. Yeh, H. R. Ke, W. P. Yang and I. H. Meng, Text summarization using a trainable summarizer and latent semantic analysis, *Information processing & management* **41**(1) (2005) 75-95.
11. D. Shen, J. T. Sun, H. Li, Q. Yang and Z. Chen, Document summarization using conditional random fields, in *IJCAI (Vol. 7)* (2007), pp. 2862-2867.
12. J. A. B. Hui Lin, 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 (ACL'11)* (2011), pp. 510-520.
13. K. Woodsend and M. Lapata, Automatic generation of story highlights, in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)* (2010), pp. 565-574.
14. G. Erkan and D. R. Radev, Lexrank: Graph-based lexical centrality as salience in text summarization, *Journal of Artificial Intelligence Research* **22** (2004) 457-479.
15. Z. Cao, F. Wei, L. Dong, S. Li and M. Zhou, Ranking with recursive neural networks and its application to multi-document summarization, in *AAAI* (2015), pp. 2153-2159.
16. Y. Zhang, M. J. Er, R. Zhao and M. Pratama, Multiview convolutional neural networks for multidocument extractive summarization, *IEEE Transactions on Cybernetics* (2016).
17. B. Z. Ramesh Nallapati, Feifei Zhai, Summarunner: A recurrent neural network based sequence model for extractive summarization of documents, in *AAAI* (2017).
18. E. Amitay and C. Paris, Automatically summarising web sites: is there a way around it?, in *Proceedings of the ninth international conference on Information and knowledge management (CIKM'00)* (2000), pp. 173-179.
19. J. Y. Delort, B. Bouchon-Meunier and M. Rifqi, Enhanced web document summarization using hyperlinks, in *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia* (2003), pp. 208-215.
20. J. T. Sun, D. Shen, H. J. Zeng, Q. Yang, Y. Lu and Z. Chen, Web-page summarization using clickthrough data, in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'05)* (2005), pp. 194-201.
21. J. Y. Delort, Identifying commented passages of documents using implicit hyperlinks, in *Proceedings of the seventeenth conference on Hypertext and hypermedia* (2006), pp. 89-98.
22. M. Hu, A. Sun and E. P. Lim, 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'08)* (2008), pp. 291-298.
23. Y. Lu, C. Zhai and N. Sundaresan, Rated aspect summarization of short comments, in *Proceedings of the 18th international conference on World wide web (WWW'09)* (2009), pp. 131-140.

24. W. Gao, P. Li and K. Darwish, 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'12)* (2012), pp. 1173-1182.
25. Y. Gong and X. Liu, 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 (SIGIR'01)* (2001), pp. 19-25.
26. C. Li, Z. Wei, Y. Liu, Y. Jin and F. Huang, Using relevant public posts to enhance news article summarization, in it COLING (2016), pp. 557-566.
27. K. M. Svore, L. Vanderwende and C. J. Burges, Enhancing single-document summarization by combining ranknet and third-party sources, in *Proceedings of EMNLP-CoNLL* (2007), pp. 448-457.
28. M. T. Nguyen, D. V. Tran, C. X. Tran and M. L. Nguyen, Learning to summarize web documents using social information, in *ICTAI* (2016), pp. 619-626.
29. M. T . Nguyen, V. D. Lai, P. K. Do, D. V . Tran and M. L. Nguyen, VSoLSCSum: Building a Vietnamese Sentence-Comment Dataset for Social Context Summarization. In *The 12th Workshop on Asian Language Resources* (2016), pp. 38-48.
30. M. T. Nguyen, C. X. Tran, D. V. Tran and M. L. Nguyen, 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'16)* (2016), pp. 2409-2412.
31. M. F. Porter, Snowball: A language for stemming algorithms (2011).
32. Y. Freund, R. D. Lyeryer, R. E. Schapire and Y. Singer, An efficient boosting algorithm for combining preferences, *Journal of Machine Learning Research* **4** (2003) 933-969.
33. F. Yoav and R. E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in *European Conference on Computational Learning Theory* (1995), pp. 23-37.
34. C. Cortes and V. Vapnik, Support-vector networks, *Machine Learning* **20**(3) (1995) 273-297.
35. T. Joachims, Training linear SVMs in linear time, in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'06)* (2006), pp. 217-226.
36. Z. Cao, T. Qin, T. Y. Liu, M. F. Tsai and H. Li, Learning to rank: from pairwise approach to listwise approach, in *Proceedings of the 24th international conference on Machine learning (ICML'07)* (2007), pp. 129-136.
37. D. M. Blei, A. Y. Ng and M. I. Jordan, Latent dirichlet allocation, *The Journal of Machine Learning Research*, **3**(Jan) (2003) 993-1022.
38. T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, in *Advances in neural information processing systems (NIPS'13)* (2013), pp. 3111-3119.
39. I. Dagan, B. Dolan, B. Magnini and D. Roth, Recognizing textual entailment: Rational, evaluation and approaches - erratum, *Natural Language Engineering* **16**(1) (2010) 105-105.
40. A. Nenkova, Automatic text summarization of newswire: lessons learned from the document understanding conference, in *AAAI (Vol. 5)* (2005), pp. 1436-1441.
41. J. D. Lafferty, A. McCallum and F. C. N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in *Proceedings of the eighteenth international conference on machine learning (Vol. 1)* (2001), pp. 282-289.
42. C. Y. Lin and E. H. Hovy, Automatic evaluation of summaries using n-gram cooccurrence statistics, in *NAACL-HLT Volume 1* (2003), pp. 71-78.