Expert Systems With Applications 76 (2017) 71–84

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Intra-relation or inter-relation?: Exploiting social information for Web document summarization☆

Minh-Tien Nguyen, Minh-Le Nguyen*

School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), 1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan

ARTICLE INFO

Article history:
Received 9 September 2016
Revised 23 January 2017
Accepted 24 January 2017
Available online 28 January 2017

Keywords:
Data mining
Information retrieval
Document summarization
Social context summarization
RTE
Ranking
Unsupervised learning

ABSTRACT

Traditional summarization methods only use the internal information of a Web document while ignoring its social information such as tweets from Twitter, which can provide a perspective viewpoint for readers towards an event. This paper proposes a framework named *SoRTESum* to take the advantages of social information such as document content reflection to extract summary sentences and social messages. In order to do that, the summarization was formulated in two steps: scoring and ranking. In the scoring step, the score of a sentence or social message is computed by using intra-relation and inter-relation which integrate the support of local and social information in a mutual reinforcement form. To calculate these relations, 16 features are proposed. After scoring, the summarization is generated by selecting top *m* ranked sentences and social messages. SoRTESum was extensively evaluated on two datasets. Promising results show that: (i) SoRTESum obtains significant improvements of ROUGE-scores over state-of-the-art baselines and competitive results with the learning to rank approach trained by RankBoost and (ii) combining intra-relation and inter-relation benefits single-document summarization.

1. Introduction

The growth of online news providers, e.g. USAToday[1], CNN[2] or Yahoo News[3] and user-generated content from social networks, e.g. tweets from Twitter[4] provides the plenty of data for users. From this, users can follow an event via the data spread. Such beneficial use is challenged by the characteristics of data explosion, e.g. diversity and noise, which people also face in extracting salient information, e.g. summary sentences in a Web document. This demands high-quality text summarization systems.

In the context of social media, readers can freely express their opinions in the form of tweets, one form of social information (Amitay & Paris, 2000; Delort, Bouchon-Meunier, & Rifqi, 2003; Hu, Sun, & Lim, 2008; Lu, Zhai, & Sundaresan, 2009; Nguyen & Nguyen, 2016; Sun et al., 2005; Wei & Gao, 2014; Yang et al., 2011), regarding a particular event. For example, after reading a Web document describing the Boston bombing event mentioned in USAToday or CNN, readers discuss the event by posting tweets on their Twitter timeline. After writing, their friends can immediately update the news and event content. Fig. 1 shows the relation of news articles and the social media. These tweets not only reveal the opinions of readers but also reflect the document content and describe the facts of an event. This observation inspires a challenging summarization task which uses the social information of a Web document to support local sentences in generating high-quality summaries.

Traditional extractive summarization methods (Edmundson, 1969; Kupiec, Pedersen, & Chen, 1995; Luhn, 1958; Osborne, 2002; Shen, Sun, Li, Yang, & Chen, 2007; Yeh, Ke, Yang, & Meng, 2005) focus on selecting summary sentences in a document by using statistical or linguistic analysis. They treat each sentence individually and train a binary classifier to classify sentences into summary (labeled by 1) or non-summary class (denoted by 0). Although these methods have achieved promising results, they only consider the internal information of a document, e.g. sentence or word/phrase level while ignoring its social information, which can provide additional information from social users, e.g. the viewpoint of users who already involve an event. The emergence of social information requires a new summary approach which exploits such kind of data to support and enrich the summarization.

This research proposes a framework which automatically extracts summary sentences and tweets of a Web document by incorporating its social information. In order to do that, we present the

☆ This manuscript is an improved and extended version of the paper: SoRTESum: A Social Context Framework for Single-Document Summarization, presented at *European Conference on Information Retrieval* (ECIR) 2016, Padova, Italy.

* Corresponding author.
E-mail addresses: tiennm@jaist.ac.jp (M.-T. Nguyen), nguyenml@jaist.ac.jp (M.-L. Nguyen).

[1] http://www.usatoday.com.
[2] http://edition.cnn.com.
[3] https://www.yahoo.com/news/.
[4] http://twitter.com - a microblogging system.

http://dx.doi.org/10.1016/j.eswa.2017.01.023
0957-4174/

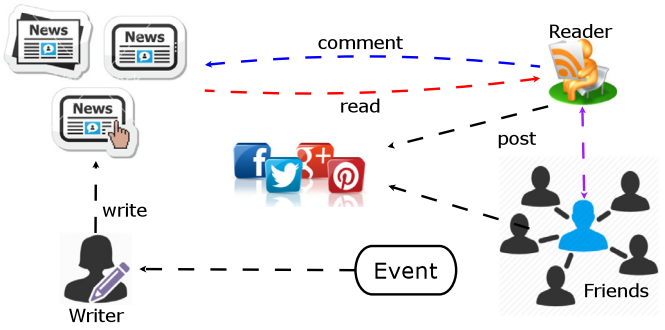


Fig. 1. A generic scheme of the relation between news and social media.

summary process in two steps namely scoring and ranking. In the scoring step, we formulate a sentence-tweet relation by using recognizing textual entailment (RTE) (a task which decides whether the meaning of a text can be plausibly inferred from another text in the same context (Dagan, Dolan, Magnini, & Roth, 2010)). Next, we model sentences and tweets in a Dual Wing Entailment Graph (DWEG), which represents the sentence-tweet entailment relation to calculate the similarity of each sentence or tweet in a mutual reinforcement fashion. We define 16 features to model the entailment relation. After modeling, each sentence or tweet is assigned a similarity score represented by two parts: intra-relation and inter-relation, which exploit the support of both local and social information. In this view, we consider sentences as the social information when modeling a tweet. In the ranking step, the framework selects top m ranked sentences and tweets, which have the highest scores as the summarization. This paper makes the following contributions:

- It proposes to define the sentence-tweet relation in the form of RTE. The relationship is different from Wei and Gao (2014); Yang et al. (2011) and Wei and Gao (2015). To the best of our knowledge, no existing methods address social context summarization by using RTE.
- It conducts a careful investigation to extract 16 RTE features represented in the form of three groups: distance, statistical, and semantic features. The investigation provides an overview of feature selection in using RTE features. The architecture of our framework is straightforward to integrate any additional features.
- It releases an open-domain dataset[5] which contains news articles along with their comments. Annotators involve in creating gold-standard references used to automatically evaluate the performance of summary methods in social context summarization. Our dataset contributes the social context summarization as well as traditional summarization.
- It proposes a unified framework[6] which utilizes 16 RTE features for calculating sentence or tweet similarity. Our method is completely unsupervised learning (scoring-then-ranking) where the input is a set of words; therefore, the framework can be applied to any domains without using external resources, e.g. syntactic parser or knowledge bases.

In remaining sections, we first introduce related works. Next, we describe SoRTESum, which uses the intra-relation and inter-relation to calculate the score of each sentence and tweet. This section also mentions our idea and model. Subsequently, we describe data collection used for our method. After preparing the data, we illustrate the summary process to achieve our goal in three steps: feature extraction, sentence scoring, and summarization. After generating the summarization, we show experimental results along with discussions and deep analyses. We finish by drawing important conclusions.

2. Literature review

2.1. Extractive summarization

Text summarization has receipted lots of attention from scientists. To the best our understanding, Luhn (1958) and Edmundson (1969) were the two first researchers who stated the text summarization task. The authors extract summary sentences by using the significant components of sentences, e.g. high-frequency content words, pragmatic words (cue words), or sentence location. In the last decade, many researchers have applied machine learning to text summarization (Kupiec et al., 1995; Osborne, 2002; Shen et al., 2007; Yeh et al., 2005). The authors define the summarization as an extraction task and represent this task as a binary classification problem, in which label 1 denotes summary, and 0 represents non-summary sentences. For example, Kupiec et al. (1995) trained a summarizer by using Bayes classifier with a set of features, e.g. sentence length, thematic words, or paragraph features. Yeh et al. (2005) used classification and latent semantic analysis (LSA) to build summarizers. The first approach uses a set of features, in which it ranks sentence position to emphasize the difference of sentences and trains the score function by using a genetic algorithm. The second method utilizes the semantic matrix of a document to generate the summarization. The highest F-score of these methods is 0.49 with 30% compression. Shen et al. (2007) exploited the sequence aspect in a document by proposing a set of features, e.g. Cosine similarity of a sentence with previous or next sentences (N = 1,2,3) and used Conditional Random Fields (Lafferty, McCallum, & Pereira, 2001) to train a classifier for selecting summary sentences. This method achieves 0.483 in ROUGE-2[7] (Recall-Oriented Understudy for Gisting Evaluation: 1-gram or 2-gram based co-occurrence statistics of output summary and gold-standard references) and 0.419 in F-1 on DUC 2001 dataset.

Lin and Bilmes (2011) designed a set of functions for document summarization. Each function guarantees two aspects: the representative and diversity. By using monotone nondecreasing and submodular functions, this method obtains the best results in term of Recall and F-score over DUC 2004 to 2007. Woodsend and Lapata (2010, 2012) formulated summary sentences in the form of concepts and defined an objective function with a set of constraints for generating highlights of a Web document. The authors represent the concepts as phrases generated from dependency trees. This method achieves 0.25 in both ROUGE-1 and F-score. There are also many studies which focus on extracting summary sentences in a document such as LexRank (Erkan & Radev, 2004), TextRank (Mihalcea & Tarau, 2004), or deep learning (Cao, Wei, Dong, Li, & Zhou, 2015; Nallapati, Zhai, & Zhou, 2016; Zhang, Er, Zhao, & Pratama, 2016).

In clustering and summarizing tweets, Wang, Shou, Chen, Chen, and Mehrotra (2015) proposed *Sumblr* to continuously summarize tweet streams. The model contains three modules: tweet clustering, online and historical summarization, and topic detection. Experimental results on large-scale tweets demonstrate the efficiency and effectiveness of this approach. Yajuan, Zhimin, Furu, Ming, and Shum (2012) exploited social influence from users and content quality to rank tweets for topic summarization. The authors first segment a tweet stream into sub-topics and then compute the

[5] Download at: http://150.65.242.101:9292/yahoo-news.zip.
[6] http://150.65.242.101:9293.
[7] https://en.wikipedia.org/wiki/ROUGE_(metric).

tweet score by integrating user information and tweet content. The high-quality summaries are decided by a classifier. The model obtains 0.4167 in ROUGE-1 over an earthquake dataset. In disaster domain, Nguyen, Kitamoto, and Nguyen (2015) proposed a ranking framework which extracts valuable tweets for reaction. The authors first cluster tweets by using topic model and then extract event in each tweet. Finally, the authors build a similarity graph and apply PageRank (Brin & Page, 1998) to select summary tweets. Results on a tornado dataset show the efficiency of exploiting event extraction for summarization.

2.2. Social context summarization

Using the support from social media for summarization has been previously studied by various approaches based on different kinds of social information such as hyperlinks (Amitay & Paris, 2000; Delort et al., 2003), click-though data (Sun et al., 2005), comments (Delort, 2006; Hu, Sun, & Lim, 2007, 2008; Lu et al., 2009), opinionated texts (Ganesan, Zhai, & Han, 2010; Kim & Zhai, 2009; Paul, Zhai, & Girju, 2010), or tweets (Gao, Li, & Darwish, 2012; Nguyen & Nguyen, 2016; Wei & Gao, 2014, 2015; Yang et al., 2011). As far as we know, Amitay and Paris (2000) were the first researchers who picked sentences from the hyperlinks of a Web document as the summarization. The authors build InCommonSense, which contains hypertext retrieval and description selection (using classification). The authors use the human to evaluate the output of InCommonSense compared again search engine results. The mean (the average values from 1 to 5 rated by users) of InCommonSense is 4.71 compared to 4.14 of AltaVisata-style (top X words from the document), and 4.13 of Google-style (query terms are highlighted, and surrounding context is taken). The summarization, however, is short because the method only selects one-sentence from linked texts as the summarization. Late, Delort et al. (2003) considered whole linked documents as the context of a Web document instead of using paragraphs including hyperlinks as Amitay and Paris (2000). The authors propose two context summarization algorithms based on similarity measurements. The first method combines both the content and context of a document and the second only takes the context. The best result is around 0.45 in term of similarity in the content and context summarization. However, similar to Amitay and Paris (2000), the authors extract the summarization from the context segments; therefore, it may not completely capture the content of a Web document compared to internal sentences.

Sun et al. (2005) addressed the problems of Amitay and Paris (2000) and Delort et al. (2003) by proposing a system which uses the help from click-through data retrieved from search engines to extract salient sentences in a Web document. This study bases on the assumption that query keywords from users typed on search engines usually reflect the content of a Web document. From this, the authors propose two methods using an adaptation of significant words (Luhn, 1958) and latent semantic analysis (Gong & Liu, 2001). The ROUGE-1 is around 0.55 in DAT1 and 0.20 in DAT2 (the two datasets used to evaluate the model). This method, however, faces two challenging issues: (i) there are no links from a new Web page to the older ones and (ii) pages which Web users click on may be irrelevant to their interest.

User-generated content such as comments was also used to support sentences for generating the summarization. Delort (2006) clustered comments by using feature vectors and selected summary sentences based on the link of vectors with the clusters. This method achieves around 50% extracted matches corresponding to post summaries. However, it requires human involvement to determine cluster relevance with summary sentences. Hu et al. (2007, 2008) extracted representative sentences that best represent topics discussed among readers in blog posts. The authors first derive salient words denoted in three graphs: topic, quotation, and mention from comments. Summary sentences are next generated by calculating the distance from each sentence to the graphs. The method obtains around 0.64 in ROUGE-1 and 0.60 in NDCG[8] on their dataset. This approach, however, only picks up sentences in a blog post while ignoring relevant information from comments. Lu et al. (2009) studied the rated aspect summarization of short comments to help users for better understanding the discussions of a target object. The authors propose a model containing three steps: aspect discovery and clustering, aspect rating prediction, and representative phrase extraction. The method obtains 0.592 of precision and 0.637 of recall in the phrase extraction on their dataset. Since this approach is used to generate the summarization of a target entity, adapting this approach for Web document summarization is still an open question due to the existence of many entities in a Web document, e.g. person or organization name.

Opinionated texts were also investigated and integrated into the summary process. Kim and Zhai (2009) studied contrastive opinion summarization in which positive and negative opinionated sentences were generated from an existing opinion summarizer. The authors formulate the summarization as an optimization problem and propose two methods that rely on measuring the content and contrastive similarity of two sentences. This approach achieves around 0.55 in precision and 0.75 in aspect coverage on their dataset. Ganesan et al. (2010) proposed Opinosis, which uses a graph-based approach for abstractive opinionated text summarization. The summaries are generated by scoring various sub-paths in the graph. The ROUGE-1 of this method is 0.327 in the hotel, car, and various product review domain. In the meantime, Paul et al. (2010) summarized contrastive viewpoints in opinionated texts by proposing a two-stages multiple viewpoint-model presented as an unsupervised probabilistic method. The model scores sentence pairs from an opposite viewpoint by using Comparative LexRank algorithm. However, adapting this approach for Web-document summarization is a challenging task.

Social messages, e.g. tweets from Twitter were widely used to support sentences in generating the summarization. Yang et al. (2011) proposed a dual wing factor graph model which uses Support Vector Machines (SVM) and Conditional Random Fields (CRF) as preliminary steps for incorporating tweets into the summarization. The authors use a ranking method, which approximates an objective function to select both summary sentences and tweets. The model achieves around 0.615 in ROUGE-1 and 0.500 in ROUGE-2 on five datasets. However, the lack of high-quality annotated data challenges this method due to using SVM and CRF. Gao et al. (2012) proposed an unsupervised method which includes a cross-collection topic-aspect modeling (cc-TAM). The authors exploit 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. The ROUGE-1 is 0.55 in the document and 0.67 in tweet selection. However, this approach does not consider the human knowledge of the summarization. Wei and Gao (2014) integrated the human knowledge into the summary process by proposing 35 features represented in three groups: local sentence, local tweet, and cross features used for a learning-to-rank model in a news highlight extraction task. The model selects top m sentences and tweets after ranking. The ROUGE-1 is 0.292 and 0.295 in document and tweet summarization, respectively. However, the salient score of a sentence or tweet is computed by using the highlights, therefore, it may be unfair compared to other methods, e.g. SVM or cc-TAM. Wei and Gao (2015) addressed the issue of Wei and Gao (2014) by proposing a variation of LexRank, which uses auxiliary tweets for building a heterogeneous graph random walk (HGRW)

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

to summarize single documents. This method achieves 0.298 in ROUGE-1 when combing summary sentences and tweets. However, HGRW may be sensitive to the noise of data (Erkan & Radev, 2004) due to the usage of LexRank.

The previous methods exist three issues: (i) supervised learning approaches require annotated data which is not always available in social context summarization, (ii) unsupervised learning methods, e.g. HGRW are sensitive to data, and (iii) several methods only select sentences or social messages as the summarization. Our method addresses the three issues in three aspects: (i) we propose a scoring-then-ranking method which treats the domain specific and the lack of high-quality annotated data problem, (ii) we consider the sensitiveness of HGRW by proposing new features which integrate human knowledge into the summarization, and (iii) our method includes both sentences and tweets instead of only selecting sentences. The selected tweets help to enrich the information of the summarization which may be not available in sentences.

3. Summarization by intra-relation and inter-relation

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

3.1. Basic idea

The data observation and literature review suggested 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.

A Web document (called document) contains a set of sentences in which summary sentences include salient information. The salient information of a sentence s_i can be measured by a similarity score, e.g. Cosine with the remaining ones. In this view, a summary sentence receives a higher similarity score compared to non-summary ones. The content of a summary sentence is usually mentioned in many tweets indicating that this sentence also receives a considerable amount of attention from readers. From the observation and hypotheses, we propose to compute the similarity score of a sentence or tweet by *intra-relation* and *inter-relation*. The intra-relation models the importance of a summary sentence compared to the remaining ones in the same document and the inter-relation integrates the support from social information.

Inspired by the idea, we propose a summarization framework named SoRTESum. The framework contains a Dual Wing Entailment Graph (DWEG) for modeling the sentence-tweet relation denoted by RTE (Dagan et al., 2010; Nguyen, Ha, Nguyen, Nguyen, & Nguyen, 2015). In Fig. 2, s_i and t_j present a sentence and tweet. Lines connecting sentences (or tweets) are the intra-relation and lines connecting sentence-tweet pairs are the inter-relation. The weight of each node, e.g. 3.25 at s_1 is a score calculated by the two relations indicating its importance. In our view, we regard tweets of a document as social information when modeling a sentence. Similarly, we consider sentences as the social information when calculating the score of a tweet. After scoring and ranking, the framework selects top m ranked sentences and tweets as the summarization.

Our study is different from Yang et al. (2011): (i) our method is unsupervised (ranking) instead of using classification and (ii) we use a set of features instead of three types of sentence-tweet relation. Our approach is similar to Wei and Gao (2014) in using ranking and the dataset; however, representing a sentence-tweet pair by a set of features is a key difference. Our method calculates the intra-relation and inter-relation by a set of features instead of using IDF-modified-cosine similarity compared to Wei and Gao (2015). Our study distinguishes the traditional methods (Edmundson, 1969; Kupiec et al., 1995; Lin & Bilmes, 2011; Luhn, 1958; Osborne, 2002; Shen et al., 2007; Yeh et al., 2005) in two aspects: (i) integrating social information and (ii) selecting both summary sentences and tweets as the summarization.

3.2. Dataset

DUC[9] 2001, 2002, and 2004 are standard datasets for text summarization; these datasets, however, lack social information. We, therefore, prepared two datasets: (i) a news highlight extraction dataset derived from Wei and Gao (2014) and (ii) our dataset collected from Yahoo News.

3.2.1. USAToday-CNN dataset

We used a news highlight extraction dataset[10] from Wei and Gao (2014). The dataset contains 121 events, 455 highlights and 78,419 tweets in 17 salient news events taken in two recent years from USAToday[11] and CNN.[12] Table 1 presents the statistics of this dataset.

Table 1 indicates that sentences and tweets share common words and the number of tweets is large enough to support sentences in generating the summarization.

3.2.2. Yahoo-News dataset

Since the USAToday-CNN dataset has no labels, training a supervised learning method, e.g. SVM or CRF is challenging. Therefore, we created a new dataset by crawling up-to-date news articles from Yahoo News[13] in May 2015. The dataset contains 157 open-domain news articles along with 3462 sentences, 5858 gold-standard references, and 25,633 comments. We asked two annotators to annotate this dataset in two rounds. In the first round, each annotator reads a complete article and selects summary sentences. After that, the annotator also reads all comments and picks up representative comments. The summary sentences and representative comments (called instances) are sentences which mainly reflect the content of a Web document. A selected instance would become a standard reference 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. Maximal selected sentences (combining both sentences and comments) are less than 35 for each document.

In the second round, the annotated data was cross-checked to show inter-annotator agreement between the two annotators. Each annotator was asked to vote on the data extracted from the other annotator. In voting, given an annotated sentence, if an annotator agrees with the pre-voted label, this sentence was also labeled by 1 and called by completely matched; otherwise, it was labeled by

[9] http://duc.nist.gov/data.html.
[10] http://www1.se.cuhk.edu.hk/~zywei/data/hilightextraction.zip.
[11] http://www.usatoday.com.
[12] http://edition.cnn.com.
[13] https://www.yahoo.com/news/.

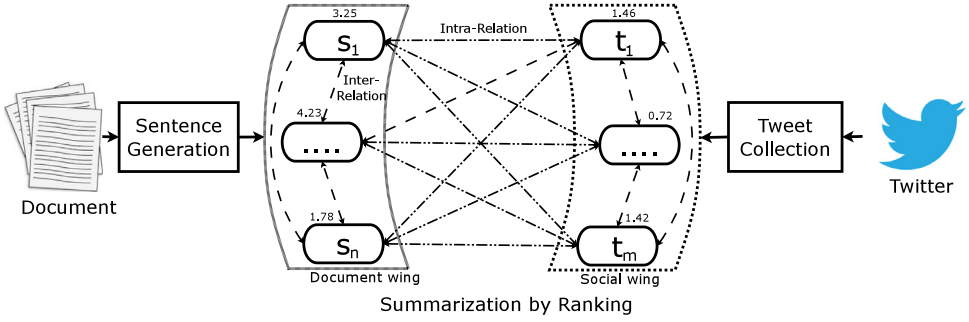


Fig. 2. The overview of summarization using intra-relation and inter-relation.

Table 1
Statistical observation was taken from Wei and Gao (2014); *s*: sentence and *t*: tweet. Note that we observed to generate results of two last rows.

	Documents	Highlights	Tweets
# Total	121	455	78,419
# Sentence per news	53.6 ± 25.6	3.7 ± 0.4	648 ± 1,161.7
# Token per news	1123.0 ± 495.8	49.6 ± 10.0	10,364.5 ± 24,749.2
# Token per sentence	21.0 ± 11.6	13.2 ± 3.2	16.0 ± 5.3
% Token overlapping	s/t: 22.24	–	t/s: 16.94
% Token overlapping (stopword removal)	s/t: 15.61	–	t/s: 12.62

Table 2
Statistical observation; *s*: sentences, *c*: comments.

Documents	Sentences	References	Comments
157	3462	5858	25,633
# Tokens	78,634	116,845	375,836
# Avg-sentences/news	22.05	37.31	163.26
# Avg-tokens/news	500.85	744.23	2393.85
# Avg-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 (no stopwords)	s/c: 8.90	–	c/s: 31.21

0. Finally, we computed the inter-annotator agreement by dividing the completely matched sentences by the entire extracted sentences. Eq. (1) defines the inter-annotator agreement.

$$agreement = \frac{\#matched\ sentences}{\#extracted\ sentences} * 100 \quad (1)$$

where: *#matched sentences* are the number of sentences which the two annotators agree with label 1; *#extracted sentences* are the number of extracted sentences corresponding to each annotator. The inter-agreement is 74.5%. We also computed Cohen's Kappa[14] (Cohen, 1968) between the two annotators. The Kappa agreement is 0.5845 indicating that the agreement is moderate based on a theoretical basis (Landis & Koch, 1977). The annotation was conducted in 75 days. This dataset was also used in Nguyen, Tran, Tran, and Nguyen (2016).

We conducted a word overlapping observation between sentences and tweets or comments. We considered each token as a single word and segmented all sentences and tweets (for USAToday-CNN dataset) and all sentences and comments (for Yahoo-News dataset). We counted the word overlapping of sentences over tweets or comments and vice versa. From Tables 1 and 2, we observe: (i) the number of two last rows indicates that 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, i.e. 22.24% of word overlapping in Table 1 and 31.21% in Table 2.

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

3.2.3. Data preparation

We removed tweets and comments with fewer than five tokens because they are too short for summarization. We used Simpson in Eq. (2) to remove near-duplicate tweets (those containing a similar content).

$$simp(t_1, t_2) = 1 - \frac{|S(t_1) \cap S(t_2)|}{min(|S(t_1)|, |S(t_2)|)} \quad (2)$$

where: *S(t)* denotes the set of words in tweet *t*. We empirically chose the similarity threshold = 0.25 by using a selection method in Section 4.4.

We used 5-fold cross-validation for USAToday-CNN dataset (the same setting with Wei & Gao (2014)) and 10-fold cross-validation for Yahoo-News dataset. We selected $m = 4$ for the first dataset because each document has 3-4 highlights and $m = 6$ for the second dataset because the number of summaries should be less than 30% average sentences per document (see Table 2). We removed stop words, hashtags, and links. Summary sentences and gold-standard references (including the highlights in the USAToday-CNN dataset) were also stemmed[15] by using the method of Porter (2011).

3.3. Summarization by SoRTESum

This section describes our process to generate the summarization in three steps: feature extraction, sentence scoring, and summarization.

3.3.1. Feature extraction

To integrate social information into the summary process, 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-tweet pair. For example, a summary sentence and tweet may not share common words (due to content variation), which may negatively affect the Cosine calculation. Therefore, we propose a set of RTE features[16] represented in the

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

[16] The RTE term was kept instead of similarity because all features were derived from RTE.

Table 3
The proposed features; *italic* in the third column denotes distance features; s_i is a sentence, t_j is a tweet; LCS is the longest common sub string.

Distance	Statistics	Semantics
Manhattan	LCS (s_i, t_j)	Semantic-sim (s_i, t_j)
Euclidean	Inclusion-exclusion	–
Cosine	% words of S in T (p(s_i, t_j))	–
Word matching (wmc)	% words of T in S (p(t_j, s_i))	*Levenshtein*
Dice coefficient	Word overlap coefficient (woc)	*JaroWinkler*
Jaccard coefficient	Smith Waterman (s_i, t_j)	*Damerau-Levenshtein*

form of three groups: distance, statistical, and semantic features to calculate the similarity between sentences and tweets. Table 3 shows our features.

Distance features: cover the distance aspect of a sentence-tweet pair, showing that a summary sentence should be close to a summary tweet rather than the meaningless ones. We consider the word and character level of a sentence-tweet 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. Eqs. (3)–(5) define the Manhattan, Euclidean, and Cosine.

$$manhattan(\vec{x},\vec{y}) = \sum_{i=1}^{n} \mid x_i - y_i \mid; \tag{3}$$

$$euclidean(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{4}$$

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

where: n is cardinality common words appearing in s_i and t_j; x_i and y_i are the frequency of each word in s_i and t_j; $\vec{x}$ and $\vec{y}$ are two same size vectors.

Eq. (6) represents the word matching coefficient.

$$wmc(s_i, t_j) = comWord(s_i, t_j) \tag{6}$$

where: *comWord()* returns the number of common words between s_i and t_j.

Eqs. (7) and (8) define the Dice and Jaccard distance.

$$dice = \frac{2. \mid X \cap Y \mid}{\mid X + Y \mid}; \tag{7}$$

$$jaccard = \frac{\mid X \cap Y \mid}{\mid X \cup Y \mid} \tag{8}$$

where: X is a set of words in s_i; and Y is a set of words in t_j.

Eq. (9) denotes the JaroWinkler distance of two texts.

$$d_j(s_i, t_j) = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_i|} + \frac{m}{|t_j|} + \frac{m-t}{m}\right) & otherwise \end{cases} \tag{9}$$

where: $|s_i|$ and $|t_j|$ are the number of characters in s_i and t_j, m is the number of matching characters and t is a half number of transpositions.

Suppose s_i can be represented by a and t_j can be denoted by b, Eq. (10) defines the DamerauLevenshtein distance.

$$d_{a,b}(i,j) = \begin{cases} max(i,j) & \text{if } min(i,j) = 0 \\ min\begin{cases} d_{a,b}(i-1,j)+1 \\ d_{a,b}(i,j-1)+1 \\ d_{a,b}(i-1,j-1)+1_{(a_i \# b_j)} \\ d_{a,b}(i-2,j-2)+1 \end{cases} & \text{if } i,j > 1 \text{ and } a_i = b_{j-1} \text{ and } a_{i-1} = b_j \\ min\begin{cases} d_{a,b}(i-1,j)+1 \\ d_{a,b}(i,j-1)+1 \\ d_{a,b}(i-1,j-1)+1_{(a_i \# b_j)} \end{cases} & otherwise \end{cases} \tag{10}$$

where: $1_{(a_i \# b_j)}$ equals 0 if $a_i = b_j$ or equals 1, otherwise. $d_{a,b}(i-1, j)+1$ is the deletion from a to b; $d_{a,b}(i, j-1)+1$ is the insertion from a to b; $d_{a,b}(i-1, j-1)+1_{(a_i \# b_j)}$ corresponds to a match or mismatch, depending on whether respective symbols are the same; $d_{a,b}(i-2, j-2)+1$ corresponds to a transposition between two successive symbols.

Eq. (11) presents the Levenshtein distance.[17]

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & \text{if } min(i,j)=0 \\ min\begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \# b_j)} \end{cases} & otherwise \end{cases} \tag{11}$$

where: $1_{(a_i \# b_j)}$ equals 0 if $a_i = b_j$ or equals 1, otherwise. $lev_{a,b}(i, j)$ is the distance between the first i characters of a and the first j characters of b.

Statistical features: capture the word overlapping aspect between a sentence and a tweet. A summary sentence and a representative tweet usually share common words (the *generation* hypothesis), indicating their content is similar. Eq. (12) represents the longest common substring of two texts.

$$lsc(s_i, t_j) = \frac{len(maxComSub(s_i, t_j))}{min(len(s_i), length(t_j))} \tag{12}$$

where: *len()* returns the length of a string; *maxComSub()* returns a number of maximum common words between s_i and t_j.

Eq. (13) shows the inclusion_exclusion coefficient.

$$\text{inclusion-exclusion}(s_i, t_j) = \frac{comWord(s_i, t_j)}{len(s_i) + len(t_j)} \tag{13}$$

where: *comWord()* returns the number of common words between s_i and t_j, *len()* returns the number of words in s_i or t_j.

Eq. (14) defines the percentage of word overlapping of s_i in t_j.

$$p(s_i, t_j) = \frac{k}{len(s_i)} \tag{14}$$

where: k is the number of common words denoted by $w = \{w_1, w_2, ..., w_k\}$ between s_i and t_j; *len()* counts the number of words in s_i. The percentage of word overlapping of t_j in s_i is also defined by changing the role of s_i and t_j.

Eq. (15) calculates the word overlap coefficient.

$$woc(s_i, t_j) = \frac{wmc(s_i, t_j)}{min(len(s_i), len(t_j))} \tag{15}$$

where: *wmc()* is the word matching coefficient of two texts s_i and t_j defined in Eq. (6), *len()* returns the number of words in a text.

The Smith-Waterman feature has been widely used in the sequence alignment, which determines similar regions between two strings. This feature compares the segments of all possible length between a sentence s_i and a tweet t_j. Eq. (16) computes this feature.

[17] This feature was used based on characters instead of words compared to Nguyen and Nguyen (2016).

$$h_{a,b}(i,j) = \begin{Bmatrix} 0 & \\ H(i-1,j-1)+s(a_j,b_j) & match/mismatch \\ max_{k\geq 1}H(i-k,j)+W_k & deletion \\ max_{l\geq 1}H(i,j-l)+W_l & insertion \end{Bmatrix} \begin{matrix} \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ 1\leq i\leq m \\ 1\leq j\leq n \end{matrix} \quad (16)$$

where: $H(i,0)=0, 0\leq i\leq m$; $H(0,j)=0, 0\leq j\leq n$; a, b are strings over s_i and t_j; m is the length of s_i; n is the length of t_j; $s(a, b)$ is a similarity function; $H(i, j)$ is the maximum similarity-score between a suffix of $a[1...i]$ and $b[1...j]$; W_i is a gap-scoring scheme.

Semantic feature: Since the distance and statistical features rely on lexical aspect; therefore, they may be limited to capture the semantics of a sentence-tweet/comment pair. For example, if a sentence contains a *"police"* word and a tweet includes an *"officer"* word, they should be closer than those which include *"police"* and *"child"*. To deal with this issue, we propose to integrate a semantic feature represented in the form of *Word2Vec* (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). The Word2Vec[18] 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.

Given the Word2Vec model, Eq. (17) calculates the semantic similarity of a sentence-tweet/comment pair.

$$sentSim(s_i,t_j) = \frac{\sum_{w_i}^{N_s}\sum_{w_j}^{N_c} w2vSim(w_i,w_j)}{N_s+N_c} \quad (17)$$

where: N_s and N_c are the number of words in s_i and t_j (stopword removal); $w2vSim()$ returns the semantic similarity between two words computed by using the *Word2Vec* model.

3.3.2. Sentence scoring

We applied the proposed features to calculate the score of each sentence and tweet in Fig. 2. More precisely, we presented two methods named *SoRTESum Iter-Wing* and *SoRTESum Dual-Wing*.

SoRTESum Inter-Wing: In this method, the framework computes the score of a sentence or a tweet by using auxiliary information from the other side. For example, the score of sentence s_i was calculated by using complementary tweets on the tweet side. Eq. (18) models the calculation.

$$score(s_i) = \frac{1}{m}\sum_{j=1}^{m} rteInterScore(s_i,t_j) \quad (18)$$

where: $s_i \in S$, $t_j \in T$, S is a set of sentences and T is a set of tweets; $rteInterScore(s_i, t_j)$ returns an entailment score between sentence s_i and tweet t_j; m is the number of tweets corresponding to each document. Eq. (19) calculates the entailment score.

$$rteInterScore(s_i,t_j) = \frac{1}{F}\sum_{k=1}^{F} f_k(s_i,t_j) \quad (19)$$

where: F contains 16 RTE features; $f()$ is a similarity function calculated by each kth feature. Similarly, the score of a tweet is also computed in the same mechanism in Eq. (20)

$$score(t_j) = \frac{1}{n}\sum_{i=1} rteInterScore(t_j,s_i) \quad (20)$$

where: n is the number of sentences in a document d.

SoRTESum Dual-Wing: In this method, the framework computes the RTE value of a sentence by using two scores: intra-score and inter-score. The intra-score captures the RTE relation of a sentence s_i with the remaining sentences in the same document and the inter-score represents the RTE relation of this sentence with auxiliary tweets. For example, the score of s_i was calculated by using s_1 to s_n; at the same time, s_i was also scored by using complementary tweets t_1 to t_m. The final value of s_i was summed by using the two scores via a balanced parameter. Eq. (21) defines the dual-wing calculation.

$$score(s_i) = \delta * \sum_{k=1}^{n} rteIntraScore(s_i,s_k) + (1-\delta) * \sum_{j=1}^{m} rteInterScore(s_i,t_j) \quad (21)$$

Similarly, the framework also computes the RTE score of a tweet with the same mechanism in Eq. (22).

$$score(t_j) = \delta * \sum_{k=1}^{m} rteIntraScore(t_j,t_k) + (1-\delta) * \sum_{i=1}^{n} rteInterScore(t_j,s_i) \quad (22)$$

where: δ is a balanced parameter which controls social information contribution in the summary process; n and m are the number of sentences and tweets. In this view, we exploit the mutual reinforcement support of sentences and tweets. Note that $rteIntraScore(s_i, t_j)$ is also computed by Eq. (19). Section 4.3 shows the selection of balanced parameter δ.

3.3.3. Summarization

The framework generates the summarization by selecting vertices which have the highest scores in the DWEG. Eq. (23) presents the selection.

$$S_r \leftarrow ranking(S); \quad T_r \leftarrow ranking(T) \quad (23)$$

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

3.4. Statistical analysis

This sections first shows methods used to compare to our framework and next presents an evaluation metric for the comparison.

3.4.1. Baseline

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

- **SentenceLead:** chooses the first m sentences as the summarization (Nenkova, 2005). This method was not used in selecting tweets or comments.
- **LexRank:** was proposed by Erkan and Radev (2004). This method builds a stochastic graph for computing the relative importance of textual units in text summarization. LexRank considers extractive text summarization by relying on the concept of sentence salience to identify the most important sentences in a document, in which the salience is typically defined by terms.

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

In this study, we applied LexRank algorithm[19] with the usage of tokenization and stemming.[20]

- **HGRW**: is a variation of LexRank named Heterogeneous Graph Random Walk (Wei & Gao, 2015). The authors utilize the help from tweets or comments to support sentences in building graphs using LexRank algorithm with the threshold is set by 0.7.
- **cc-TAM**: was proposed by Gao et al. (2012). The authors build a cross-collection topic-aspect modeling (cc-TAM) as a preliminary step to generate a bipartite graph used for co-ranking to select sentences and tweets for multi-document summarization.
- **Learning to Rank (L2R)** (Wei & Gao, 2014): The authors propose 35 features and adopt RankBoost implemented in RankLib[21] for training two L2R models using ERR metric score in 300 iterations, one for sentences and the other for tweets. In training, when modeling a sentence, the L2R model combines social features from tweets with local features of this sentence. Similarity, social features from sentences are also used to support the local features when modeling a tweet. In our study, we ignored unnecessary features, e.g. hashtags, URLs or quality-depend because they are not usually available in comments. This method contains two baselines: only using local features for sentences or tweets/comments (L2R), and combining local and cross features (CrossL2R).
- **SVM** (Cortes & Vapnik, 1995): was used by Yang et al. (2011). The authors train a binary classifier on training data and apply the classifier to testing data to create the summarization. The summarization is generated by selecting sentences or comments labeled by 1. In our study, we used LibSVM[22] with RBF kernel. Features were scaled in [−1, 1], and comments were weighted by 85% due to imbalanced data (see the Table 2). Note that this method is only used for the Yahoo-News dataset because labels are unavailable in the USAToday-CNN dataset.
- **RTE One-Wing:** uses one wing information (document or tweet/comment) to calculate the RTE score by using the proposed features. For example, the method only uses the support from the remaining sentences when calculating the score of a sentence. Similarly, the remaining tweets or comments are also utilized to compute the score of a tweet or comment.

3.4.2. Evaluation metric

In the USAToday-CNN dataset, we used the highlights as gold-standard references. In the Yahoo-News dataset, we used selected sentences and comments (those which were labeled by 1 in the annotation step) as gold-standard references. For evaluation, we employed F-1 ROUGE-N[23] (Lin & Hovy, 2003)(N = 1, 2), in which Eq. (24) defines ROUGE-N:

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

where: n is the length of n-gram, $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and the references, $Count(gram_n)$ is the number of n-grams in the references.

[19] https://code.google.com/p/louie-nlp/source/browse/trunk/louie-ml/src/main/java/org/louie/ml/lexrank/?r=10.
[20] http://nlp.stanford.edu/software/corenlp.shtml.
[21] https://people.cs.umass.edu/~vdang/ranklib.html.
[22] http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
[23] http://kavita-ganesan.com/content/rouge-2.0-documentation.

Table 4
Summary performance; * denotes supervised learning methods; **bold** is the best value; *italic* is the second best. Note that this dataset has no label, hence SVM was not used.

Method	Document		Tweet	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
Sentence Lead	0.249	0.096	–	–
LexRank	0.183	0.045	0.154	0.056
HGRW	*0.271*	0.091	0.207	0.053
cc-TAM	0.261	0.074	**0.248**	0.071
L2R*	0.248	0.086	0.199	0.064
CrossL2R*	0.270	**0.111**	0.209	0.069
RTESum One-Wing	0.202	0.072	0.191	0.067
SoRTESum Inter-Wing	**0.283**	*0.099*	*0.230*	*0.072*
SoRTESum Dual-Wing	0.209	0.055	*0.230*	**0.077**

4. Results and discussion

In order to measure our success, in Section 4.1 we first show results of SoRTESum compared to the baselines. The comparison answers two questions: (i) whether the performance of our method can compare to other methods and (ii) whether our approach is efficient. Sections 4.2–4.4 investigate feature contribution, the role of the trade-off parameter in Eqs. (21) and (22), and the relationship between tweet-size and computational time. We also observe the position distribution of summaries generated from our model. This observation reveals sentence position aspect in the summarization. We finally validate our hypotheses and deeply analyze the model by a running example.

4.1. Experimental results

We first evaluated our method on the USAToday-CNN dataset. The results in Table 4 show that SoRTESum outperforms the baselines in ROUGE-1 of document summarization and ROUGE-2 of tweet extraction, and obtains competitive results in other cases. The results support our idea and hypotheses stated in Section 3.1. The performance of document summarization is better than that in the tweet selection because tweets were usually generated from news article content (Nguyen & Nguyen, 2016; Wei & Gao, 2014; Yang et al., 2011). This aspect supports the *reflection* hypothesis.

SoRTESum Inter-Wing outperforms L2R and CrossL2R (Wei & Gao, 2014) in both ROUGE-1, 2 of document and tweet summarization, i.e. 0.283 vs. 0.270, even though the L2R is a supervised learning method. This shows the efficiency of our approach and features. In other words, our method comparably performs CrossL2R (Wei & Gao, 2014) in ROUGE-2 of sentence extraction, i.e. 0.099 vs. 0.111. This is because: (i) CrossL2R is also a supervised learning method and (ii) CrossL2R computes the salient score of a sentence or tweet by using the maximal ROUGE-1 F-score of this instance with the corresponding ground-truth highlights. As a result, this model tends to extract summaries which are highly similar to the highlights. However, even with this calculation, in ROUGE-1 of sentence selection, our model is the best.

SoRTESum Inter-Wing significantly surpasses SoRTESum Dual-Wing in document summarization, i.e. 0.283 vs. 0.209 in ROUGE-1; however, in tweet extraction, they are comparable. In the sentence selection, many tweets were derived by using the title of a Web document (readers copy the title to create their tweets); therefore, after removing near-duplicate tweets, the remaining ones contain vital information to reflect the content of a document. As a result, adding information from the document (dual wing) is unnecessary. In tweet summarization, the score of a tweet in SoRTESum Dual-Wing was calculated by an accumulative fashion; therefore, the model tends to select longer tweets compared to SoRTRSum Inter-Wing. However, the gap between the two methods is small, i.e.

Table 5
Summary performance on the Yahoo-News dataset.

Method	Document		Comment	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
Sentence Lead	0.360	*0.309*	–	–
LexRank	0.328	0.257	0.244	0.140
HGRW	0.377	0.321	0.248	*0.145*
cc-TAM	0.321	0.268	0.166	0.088
L2R*	0.353	0.307	0.205	0.098
CrossL2R*	0.363	**0.321**	0.217	0.111
SVM*	0.293	0.239	0.141	0.074
RTESum One-Wing	0.352	0.294	0.222	0.118
SoRTESum Inter-Wing	*0.395*	0.294	**0.354**	**0.164**
SoRTESum Dual-Wing	**0.397**	0.301	*0.309*	0.139

0.077 vs. 0.072 in ROUGE-2. The results of SoRTESum Inter-Wing and Dual-wing reveal that information from a document can also be exploited to support tweet summarization.

We extensively validated SoRTESum on our dataset collected from Yahoo News. In Table 5, our method is the best except for ROUGE-2 of document summarization. The comparison trend is consistent with the results in Table 4 and validates the efficiency of our method. This shows that the performance of our method can reach to supervised learning methods, e.g. L2R or SVM.

Interestingly, in Table 5, the performance of SoRTESum Dual-Wing is better than that of SoRTESum Inter-Wing in sentence extraction compared to the results in Table 4. This is because comments in the Yahoo-News dataset do not include the title of documents. In this case, the internal information of a document is useful for supporting social information to generate the summarization. Note that the performance is slightly different. In the tweet selection, SoRTESum Inter-Wing clearly outperforms SoRTESum Dual-Wing.

The results from Tables 4 and 5 indicate that SoRTESum outperforms L2R and SVM, two supervised learning methods. This shows the efficiency of our method and features. SoRTESum dominates LexRank due to the integration of social information (Nguyen & Nguyen, 2016; Wei & Gao, 2014, 2015). Our method surpasses HGRW and cc-TAM, two state-of-the-art methods in social context summarization. In the first case, in USAToday-CNN dataset, HGRW achieves the second best result in ROUGE-1 of sentence extraction. However, HGRW is a variation of LexRank, which uses *IDF-modified-cosine similarity*; therefore, the noise of data may negatively affect the summarization (Erkan & Radev, 2004). For example, HGRW is the worst in ROUGE-2 of tweet selection in Table 4, i.e. 0.053. In the second case, cc-TAM is the best in ROUGE-1 of tweet summarization in Table 4, i.e. 0.248; however, the performance of cc-TAM is quite poor in Table 5, i.e. 0.166 vs. 0.354 in ROUGE-1 of tweet summarization. This is because cc-TAM is designed for multi-document summarization, while our dataset is created for single-document summarization. SentenceLead is a strong baseline because it simulates the summarization by picking up m first sentences (Nenkova, 2005). SVM achieves quite poor results due to the feature conflict as similar as L2R. RTESum One-Wing obtains competitive results due to the usage of our features. This suggests that even with the lack of social information, our approach still comparably performs with other methods.

All the methods in Tables 4 and 5 are inefficient with informal social messages, i.e. very short, abbreviated, or ungrammatical tweets or comments. This is possibly solved by integrating a sophisticated pre-processing step. In addition, tweets or comments did not come from the news sources challenge all the approaches due to content inconsistency between sentences and social messages. This can be addressed by integrating a refined crawling method to capture relevant information from other sources. Our method is also limited if the content of sentences and tweets or comments is highly abstractive, e.g. need an inference. In this case, a more novel approach, e.g. RTE should be considered.

4.2. Feature contribution analysis

We further examined feature contribution in our model by removing each feature and keeping $n-1$ ones (leave-one-out test). Feature weight was calculated by the minus of SoRTESum Inter-Wing using all features with the model using $n-1$ features on the USAToday-CNN dataset. Table 6 presents top six effective features.

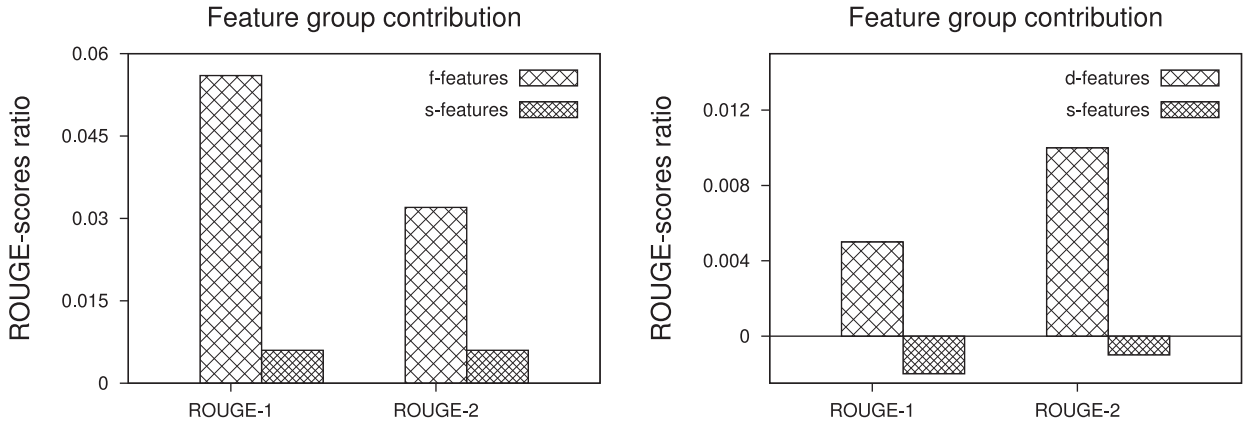
Table 6 indicates that both the distance and statistical features affect the summarization. In the sentence selection, the distance features play an important role. This shows that summary sentences include salient common words or phrases. In document and tweet summarization, Dice coefficient, Inclusion-exclusion coefficient, and Jaccard positively affect the extraction. In the tweet selection, the distance features are more important than the statistical ones (only inclusion-exclusion coefficient and matching appear). The Word2vec feature has no contribution in ROUGE-1 of sentence selection; however, in ROUGE-2 and tweet extraction, it positively contributes to the model.

We investigated the contribution of the distance (d-features) and statistical features (s-features). Since the contribution of word2vec was already shown in Table 3, we only observed the two other groups. We computed the F-1 ROUGE-scores ratio on the y-axes by the minus of SoRTESum Inter-Wing using all feature groups with the model using two other groups (removing the distance or statistical features and keeping the other ones).

Fig. 3 shows that in document summarization, both the two feature groups positively contribute to the model, in which the distance features have a bigger influence compared to the statistical ones. This trend is consistent in Fig. 3b for tweet selection. This supports the results in Table 6, in which the number of distance features is larger than that of the statistical ones, and concludes that the distance features play an important role in document and tweet summarization. In Fig. 3b, the statistical features slightly negatively affect the model because although each statistical feature in Table 6 has a positive impact, combining them may lead to feature conflict (the negative values are tiny).

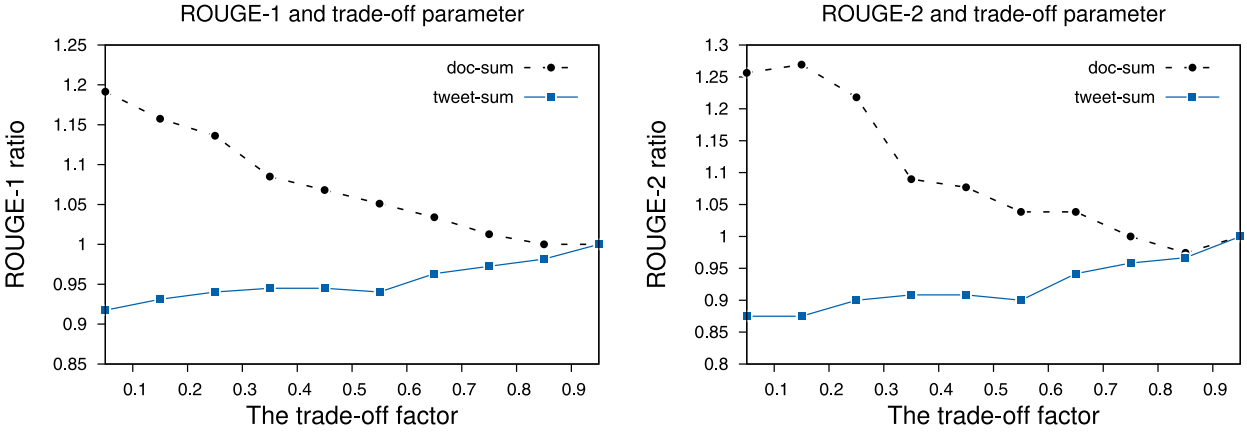
Table 6
Top six effective features generated from SoRTESum Inter-Wing on the USAToday-CNN dataset; * is Inclusion-exclusion coefficient; *italic* denotes the statistical features.

Feature	Document		Feature	Tweet	
	ROUGE-1	ROUGE-2		ROUGE-1	ROUGE-2
dice	0.1×10^{-3}	0.6×10^{-3}	dice	0.9×10^{-3}	0.3×10^{-3}
overlap	0.1×10^{-3}	0.5×10^{-3}	manhattan	0.9×10^{-3}	0.1×10^{-3}
*in-ex coeffi**	0.8×10^{-3}	0.5×10^{-3}	*in-ex coeffi**	0.9×10^{-3}	0.2×10^{-3}
jaccard	0.1×10^{-3}	0.000	jaccard	0.9×10^{-3}	$0.1 \times 10^{-}3$
cosine	0.1×10^{-3}	0.5×10^{-3}	*matching*	0.7×10^{-3}	0.000
w2v	0.000	0.2×10^{-3}	w2v	0.9×10^{-3}	0.1×10^{-3}



(a) Feature group in document (b) Feature group in tweet

Fig. 3. The contribution of feature groups in SoRTESum Inter-Wing.



(a) The performance in ROUGE-1 (b) The performance in ROUGE-2

Fig. 4. The adjustment of δ in SoRTESum Dual-Wing.

4.3. Tuning Trade-off Parameter

We investigated the impact of the balanced parameter in Eqs. (21) and (22) by adjusting δ in [0.05, 0.95] with a jumping step = 0.1. The ratio ROUGE-scores of SoRTESum Dual-Wing was computed by dividing the ROUGE-scores of each tuning point for the ROUGE-scores at $\delta = 0.95$.

Fig. 4 shows that the performance of document and tweet summarization is in an inverted form. When increasing δ, the performance of sentence extraction decreases while the performance of tweet summarization raises. Auxiliary information benefits the summary process until a turning point. When δ is closing to 0.85, the general performance of our method is improved. To balance, we empirically select $\delta = 0.85$. Note that the change is slightly different among tuning points because the score of a sentence or tweet was computed by averaging the score of RTE features; therefore, the role of δ may be saturated.

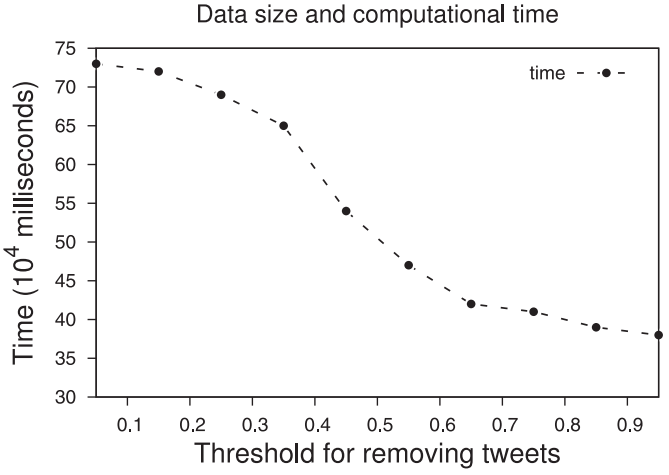
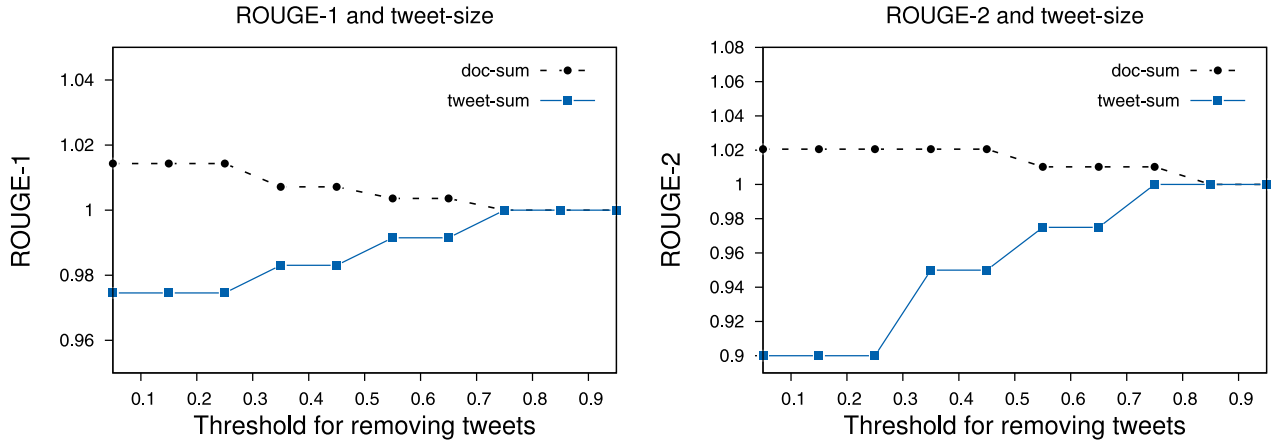


Fig. 5. The relationship between tweet-size and computational time.

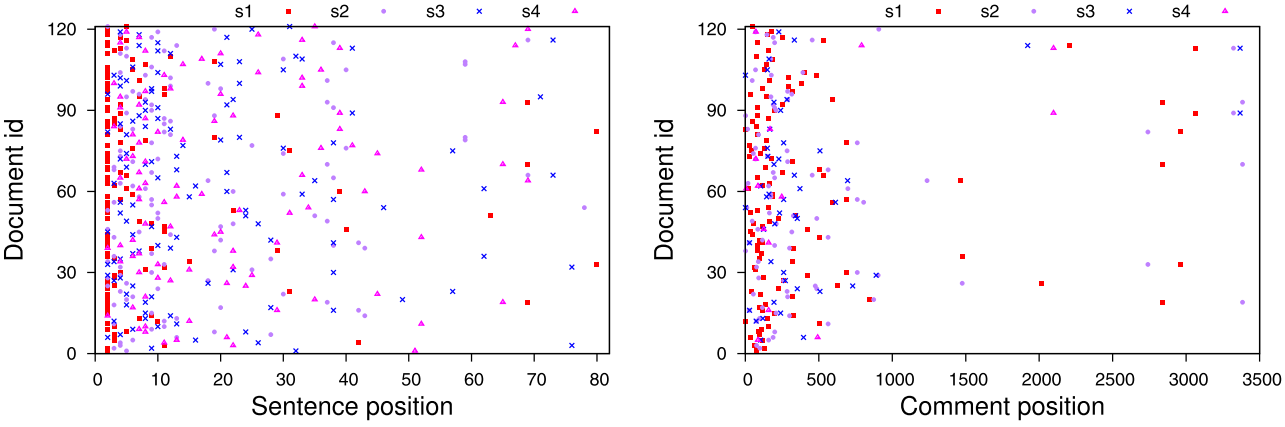
4.4. Tweet-size analysis

We also observed the impact of tweet-size in SoRTESum Inter-Wing by adjusting the similarity threshold in Eq. (2) from [0.05, 0.95] with a jumping step = 0.1 on the USAToday-CNN dataset. Fig. 5 shows that when increasing the similarity threshold (removing more near-duplicate tweets), the computational time decreases. It slightly falls from 0.05 to 0.35 and significantly decreases from 0.35 to 0.65. In the remaining points, the trend indicates that removing more tweets slightly speeds up the system.



(a) Document summarization (b) Tweet summarization

Fig. 6. The relation between ROUGE-scores and tweet-size.



(a) Document summarization (b) Tweet summarization

Fig. 7. The position of summary sentences and tweets.

We investigated the relation between data size and the summary performance of SoRTESum Inter-Wing. The ROUGE-scores ratio was computed by dividing the ROUGE-scores of each data size point for the scores at 0.95.

In Fig. 6a and b, the trend indicates that the performance of document summarization decreases while the performance of tweet extraction raises when removing near-duplicate tweets. This is because removing tweets reduces the support from the social information of sentence selection. Note that the gap among data points is small. For tweet summarization, the removal improves the summary performance. This is because SoRTESum Inter-Wing can correctly select the summaries in a smaller set. In additional, recall that many tweets were generated by using the title of a document; therefore, the removal helps to keep salient tweets, which reflect the content of the document. To balance the performance and computational time, the threshold was chosen by 0.25.

4.5. Sentence position observation

We further investigated the position of summaries generated from SoRTESum Dual-Wing on the USAToday-CNN dataset. Summary sentences and tweets were matched again to the original documents to find their positions. Fig. 7 presents the position distribution of summary sentences and tweets.

From Fig. 7, we observe that most of the summary sentences locate within first 15 sentences on the document side and first 300 tweets on the tweet side. There are also outlier points, e.g. 80th in Fig. 7a and 3500th in Fig. 7b because several documents contain a larger number of sentences and tweets (thousand tweets). Considering the data observation in Tables 1 and 2, we conclude that: (i) the density distribution of tweets is sparse because sequence aspect does not explicitly exist on the tweet side, and (ii) SentenceLead (Nenkova, 2005) is inefficient in tweet or comment summarization because representative social messages usually appear in a wider range compared to sentences.

4.6. Hypothesis analysis

We deeply analyzed our hypotheses in Section 3.1 by running an example generated from SoRTESum Dual-Wing. The example contains two sentences and tweets shown in Table 7, in which S_1 and T_2 are summary sentences and S_2 and T_1 are non-summary sentences.

Fig. 8 indicates that the summary sentences, i.e. S_1 and T_2 receive higher scores compared to non-summary sentences, i.e. S_2 and T_1. This validates our idea stated in Section 3.1. In addition, $T1$ and $T2$ contain the important information of the Boston bombing event. This supports the *representation* and *reflection* hypoth-

Table 7
An example of the Boston Bombing (24) in the USAToday-CNN dataset.

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

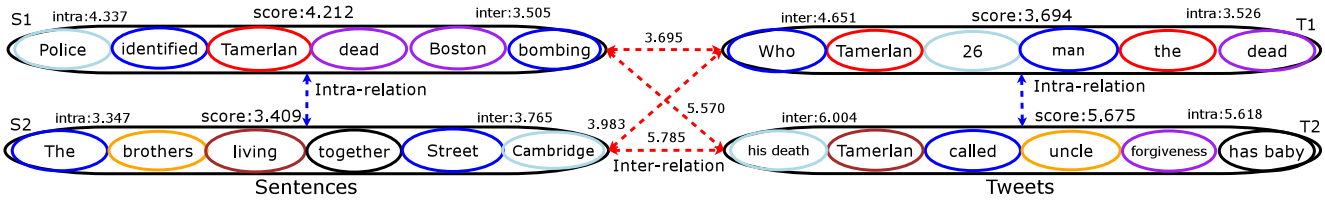


Fig. 8. A running example generated by SoRTESum Dual-Wing.

Table 8
A summary example; [+] shows a strongly relevance and [-] is slightly relevant.

Highlights	
Police identified Tamerlan Tsarnaev, 26, as the dead Boston bombing suspect. Tamerlan studied engineering at Bunker Hill Community College in Boston. He was a competitive boxer for a club named Team Lowell.	
Summary Sentences	
[+] S1: Tamerlan Tsarnaev, the 26-year-old identified by police as the dead Boston bombing suspect, called his uncle Thursday night and asked for forgiveness, the uncle said. [+] S2: Police have identified Tamerlan Tsarnaev as the dead Boston bombing suspect. [+] S3: Tamerlan attended Bunker Hill Community College as a part-time student for three semesters, Fall 2006, Spring 2007, and Fall 2008. [-] S4: He said Tamerlan has relatives in the United States and his father is in Russia.	
Summary Tweets	
SoRTESum Inter-Wing	SoRTESum Dual-Wing
[+] T1: Before his death Tamerlan Tsarnaev called an uncle and asked for his forgiveness. Saidhe is married and has a baby. [-] T2: I proudly say I was the 1st 1 to write this on twitter. Uncle, Tamerlan Tsarnaev called, asked for forgiveness. [-] T3: So apparently the dead suspect has a wife & baby? And beat his girlfriend enough to be arrested? (same woman?). [+] T4: Tamerlan Tsarnaev ID'd as dead Boston blast suspect - USA Today - USA TODAY, Tamerlan Tsarnaev ID'd as dead.	[-] T1: I proudly say I was the 1st 1 to write this on twitter. Uncle,Tamerlan Tsarnaev called, asked for forgiveness. [-] T2: So apparently the dead suspect has a wife & baby? And beat his girlfriend enough to be arrested? (same woman?). [+] T3: Before his death Tamerlan Tsarnaev called an uncle and asked for his forgiveness. Said he is married and has a baby. [+] T4: #BostonMarathon bomber Tamerlan called uncle couple of hours before he was shot dead said 'I love you and forgive me.

esis. We also observe that sentences and tweets share common words, e.g. *"Tamerlan", "bombing", "dead"* supporting the *generation* and *common topic* hypothesis.

4.7. Error analysis

We conducted an error analysis to show the limitation of our method. In Table 8 (the Web interface can be seen at SoRTESum system[24]), both the two methods yield the same output in document summarization, in which *S1, S2*, and *S3* are summary sentences. The content of these sentences completely relates to the highlights, which mention the death of Tamerlan Tsarnaev at the Boston bombing event or attending information in his college. This is because they contain important words; hence our method can select correctly. In contrast, *S4* mentioning his father information is slightly relevant.

In the tweet extraction, the two methods generate three similar tweets, and the remaining one is different. The summarization contains the same tweets, i.e. *T1* in SoRTESum Inter-Wing and *T3* in SoRTESum Dual- Wing; the other ones are different leading to the different performance between the two methods. They are quite relevant to the event but do not directly mention the death of Tamerlan Tsarnaev, e.g. *T2*. This is because *T2* also include important information which challenges our method and leads to an incorrect selection.

Irrelevant data may negatively affect the summarization because the score of a sentence or tweet was calculated by an accumulative mechanism; therefore, non-summary sentences or tweets can achieve a high score, e.g. *T2*. They do not directly mention the death of Tamerlan Tsarnaev but receives a lot of attention from readers. All sentences and tweets in Table 8 contain keywords which suggest that the summary performance can be improved based on informative phrases as the *generation* hypothesis stated in Section 3.1.

[24] http://150.65.242.101:9293.

5. Conclusion

This paper presents *SoRTESum*, a novel ranking framework which utilizes the social information of a Web document to generate a high-quality summarization. Our framework combines the intra-relation and the inter-relation to calculate the score of each sentence or tweet and ranks to select top m sentences and tweets as the summarization. This paper concludes that integrating social information and formulating a sentence-tweet pair by a set of features benefit the sentence selection. In the first aspect, social information supports local information to improve the quality of the summary process. The social information not only comes from tweets or comments but also derives from sentences due to the mutual reinforcement support. In the second aspect, the feature combination helps to capture various similarity aspects. We extensively validate our method on a news highlight extraction dataset taken from USAToday and CNN and our released dataset collected from Yahoo News. Experimental results show that SoRTESum achieves improvements over state-of-the-art baselines and our features are efficient for single-document summarization.

For future directions, other important features of the RTE task, e.g. named entity recognition or tree edit distance should be considered and integrated into the model. Our problem should also be represented in a deeper model, e.g. LSTM or CNN to enrich the semantic aspect. To ensure the quality of the summarization, human evaluation should also be considered.

Acknowledgment

This work was supported by JSPS KAKENHI Grant number 15K16048, JSPS KAKENHI Grant Number JP15K12094, and CREST, JST. We would like to thank Preslav Nakov and Wei Gao for useful discussions and insightful comments on earlier drafts; Chien-Xuan Tran for building the Web interface. We also thank the comments of anonymous reviewers for improving our paper.

References

Amitay, E., & Paris, C. (2000). Automatically summarising web sites: Is there a way around it. In *Proceedings of the ninth international conference on information and knowledge management (CIKM)* (pp. 173–179). ACM.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN system, 30*(1), 107–117.

Cao, Z., Wei, F., Dong, L., Li, S., & Zhou, M. (2015). Ranking with recursive neural networks and its application to multi-document summarization. In *AAAI* (pp. 2153–2159). ACM.

Cohen, J. (1968). Weighed kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin, 70*(4), 213–220.

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

Dagan, I., Dolan, B., Magnini, B., & Roth, D. (2010). Recognizing textual entailment: Rational, evaluation and approaches - erratum. *Natural Language Engineering, 16*(1), 105.

Delort, J. Y. (2006). Identifying commented passages of documents using implicit hyperlinks. In *Proceedings of the seventeenth conference on hypertext and hypermedia* (pp. 89–98). ACM.

Delort, J. Y., Bouchon-Meunier, B., & Rifqi, M. (2003). Enhanced Web document summarization using hyperlinks. In *Proceedings of the fourteenth ACM conference on hypertext and hypermedia* (pp. 208–215). ACM.

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the Association for Computing Machinery (JACM), 16*(2), 264–285.

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

Ganesan, K., Zhai, C., & Han, J. (2010). Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics (COLING)* (pp. 340–348). Association for Computational Linguistics.

Gao, W., Li, P., & Darwish, K. (2012). Joint topic modeling for event summarization across news and social media streams. In *Proceedings of the 21st ACM international conference on information and knowledge management (CIKM)* (pp. 1173–1182). ACM.

Gong, Y., & Liu, X. (2001). Generic text summarization using relevant measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 19–25). ACM.

Hu, M., Sun, A., & Lim, E. P. (2007). Comments-oriented blog summarization by sentence extraction. In *Proceedings of the sixteenth ACM conference on information and knowledge management (CIKM)* (pp. 901–904). ACM.

Hu, M., Sun, A., & Lim, E. P. (2008). Comments-oriented document summarization: Understanding document with readers' feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 291–298). ACM.

Kim, H. D., & Zhai, C. (2009). Generating comparative summaries of contradictory opinions in text. In *Proceedings of the 18th ACM conference on information and knowledge management (CIKM)* (pp. 385–394). ACM.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 68–73). ACM.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *In proceedings of the eighteenth international conference on machine learning: Vol. 1* (pp. 282–289). ICML. June.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics, 33*, 159–174.

Lin, C. Y., & Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology: Vol.1* (pp. 71–78). Association for Computational Linguistics.

Lin, H., & Bilmes, J. (2011). A class of submodular functions for document summarization. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies: Vol.1* (pp. 510–520). Association for Computational Linguistics.

Lu, Y., Zhai, C., & Sundaresan, N. (2009). Rated aspect summarization of short comments. In *Proceedings of the 18th international conference on world wide web (WWW)* (pp. 131–140). ACM.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research Development, 2*(2), 159–165.

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems (NIPS)*, 3111–3119.

Nallapati, R., Zhai, F., & Zhou, B. (2016). SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents, AAAI 2017.

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

Nguyen, M. T., Ha, Q. T., Nguyen, T. D., Nguyen, T. T., & Nguyen, L. M. (2015). Recognizing textual entailment in vietnamese text: An experimental study. In *The seventh international conference on knowledge and systems engineering (KSE)* (pp. 108–113). IEEE.

Nguyen, M. T., Kitamoto, A., & Nguyen, T. T. (2015). TSum4act: A framework for retrieving and summarizing actionable tweets during a disaster for reaction. In *Pacific-asia conference on knowledge discovery and data mining (PAKDD)* (pp. 64–75). Springer International Publishing.

Nguyen, M. T., & Nguyen, M. L. (2016). SoRTESum: A social context framework for single-document summarization. In *European conference on information retrieval (ECIR)* (pp. 3–14). Springer International Publishing.

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

Osborne, M. (2002). Using maximum entropy for sentence extraction. In *Proceedings of the ACL-02 workshop on automatic summarization: Vol.4* (pp. 1–8). Association for Computational Linguistics.

Paul, M. J., Zhai, C., & Girju, R. (2010). Summarizing contrastive viewpoints in opinionated text. In *Proceedings of the 2010 conference on empirical methods in natural language processing (EMNLP)* (pp. 66–76). Association for Computational Linguistics.

Porter, M. F. (2011). Snowball: A language for stemming algorithms.

Shen, D., Sun, J. T., Li, H., Yang, Q., & Chen, Z. (2007). Document summarization using conditional random fields. *IJCAI, Vol.7*, 2862–2867.

Sun, J. T., Shen, D., Zeng, H. J., Yang, Q., Lu, Y., & Chen, Z. (2005). Web-page summarization using clickthrough data. In *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 194–201). ACM.

Wang, Z., Shou, L., Chen, K., Chen, G., & Mehrotra, S. (2015). Text summarization using a trainable summarizer and latent semantic analysis. *IEEE Transactions on Knowledge and Data Engineering, 27*(5), 1301–1315.

Wei, Z., & Gao, W. (2014). Utilizing microblogs for automatic news highlights extraction. In *COLING* (pp. 872–883). Association for Computational Linguistics.

Wei, Z., & Gao, W. (2015). Gibberish, assistant, or master?: Using tweets linking to news for extractive single-document summarization. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval* (pp. 1003–1006). ACM.

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

Woodsend, K., & Lapata, M. (2012). Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-coNLL)* (pp. 233–243). Association for Computational Linguistics. July.

Yajuan, D., Zhimin, C., Furu, W., Ming, Z., & Shum, H. Y. (2012). Twitter topic summarization by ranking tweets using social influence and content quality. In *Proceedings of the 24th international conference on computational linguistics (COLING)* (pp. 763–780). Association for Computational Linguistics.

Yang, Z., Cai, K., Tang, J., Zhang, L., Su, Z., & Li, J. (2011). Social context summarization. In *Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval* (pp. 255–264). ACM.

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

Zhang, Y., Er, M. J., Zhao, R., & Pratama, M. (2016). Multiview convolutional neural networks for multidocument extractive summarization. *IEEE Transactions on Cybernetics*.