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.

Abstract

Traditional summarization methods only use inherent information of a Web document while ignoring its social information such as tweets from Twitter, which can provide a perspective viewpoint for readers towards a special event. This paper proposes a framework named *SoRTESum* to take the advantage of social information such as document content reflection to extract important sentences and social messages as the summarization. In order to to 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, 14 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 indicate that: (i) SoRTESum obtains significant improvements of ROUGE-score over state-of-the-art methods and competitive results with learning to rank trained by RankBoost and (ii) combining intra and inter-relation benefits single-document summarization.

☆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: Minh-Le Nguyen

Email addresses: tiennm@jaist.ac.jp (Minh-Tien Nguyen), nguyenml@jaist.ac.jp (Minh-Le Nguyen)

Keywords: Data Mining, Information Retrieval, Document Summarization, Social Context Summarization, RTE, Ranking, Unsupervised Learning.

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. Twitter[4] provides 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. important 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 (Nguyen and Nguyen, 2016; Amitay and Paris, 2000; Delort et al., 2003; Sun et al., 2005; Hu et al., 2008; Lu et al., 2009; Yang et al., 2011; Wei and Gao, 2014), regarding a special event mentioned in a Web document. For example, after reading a Web document describing the Boston bombing event in USAToday or CNN, readers talk about the event by posting tweets on their Twitter timeline. After writing, their friends can immediately update the news content. The relation of news article and social media is shown in Figure 1. These tweets not only reveal the opinions of readers but also reflect the content of a document and describe the facts of an event. This inspires a novel summarization task which uses the social information of a Web document to support local sentences in generating the summarization.

Traditional extractive summarization methods (Luhn, 1958; Edmundson, 1969; Kupiec et al., 1995; Osborne, 2002; Yeh et al., 2005; Shen et al., 2007) focus on selecting important sentences in a document by using statistical or

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

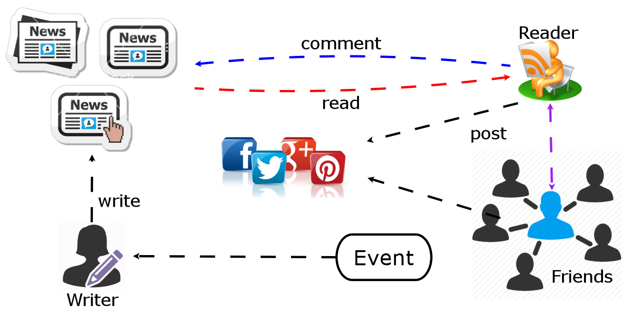


Figure 1: A generic scheme of the relation between news and social media

linguistic information. They treat each sentence individually and train a binary classifier to classify sentences, in which label 1 denotes summary sentences and 0 represents non-summary sentences. Although these methods have achieved promising results, they only consider inherent document information, 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. This demands a new summary approach which integrates social information to enrich the summarization.

This research aims to propose a summary model which automatically extracts important sentences and representative tweets of a Web document by incorporating its social information. In order to do that, the summarization was formulated in two steps namely scoring and ranking. In the scoring step, a sentence-tweet relation was formulated by recognizing textual entailment (RTE) in which social information from tweets was utilized to support local information in sentences. Next, sentences and tweets were modeled in a dual wing entailment graph (DWEG), which represents the sentence-tweet entailment relation to calculate the similarity of each sentence or tweet based on mutual reinforcement information. The entailment relation was represented by 14 features. After modeling, the similarity score of each sentence or tweet was formulated by two parts: intra-relation and inter-relation, which exploit the support of both local and social information. In this view, sentences were also deemed as social information when modeling a tweet. In the ranking step, top m ranked sentences and tweets were selected as the summarization. This paper makes the following contributions:

- It proposes to formally define sentence-tweet relation in the form of recognizing textual entailment (RTE). The relation is different to (Yang et al., 2011; Wei and Gao, 2014; 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 14 RTE features represented in the form of two groups: distance and statistical features. The investigation provides an overview of feature selection in using RTE features. The architecture of our model is straightforward to integrate any additional features.
- It releases an open-domain dataset[5] which contains news articles along with their comments. The standard selected sentences and comments are used to automatically evaluate the performance of summary systems in social context summarization. Our dataset contributes social context summarization as well as traditional summarization.
- It proposes a unified framework[6] which utilizes 14 RTE features for calculating sentence or tweet similarity. Our method is completely unsupervised learning (scoring-then-ranking) with input is words; therefore, the framework can be applied for unrestricted 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 intra-relation and inter-relation to calculate the score of each sentence and tweet. Our idea along with SoRTESum model are also mentioned this section. Subsequently, we describe data collection used for our model. After preparing the data, we illustrate our process to achieve our goal in three steps: feature extraction, calculation modeling, and summarization.

[5] Download at: http://150.65.242.101:9292/yahoo-news.zip
[6] http://150.65.242.101:9293

After generating the summarization, we show experimental results along with discussions and deep analyses. We finish by drawing important conclusions.

2. Literature Review

Using the support from social media for summarization has been previously studied by various approaches based on different kind of social information such as hyperlinks (Amitay and Paris, 2000; Delort et al., 2003), click-though data (Sun et al., 2005), comments (Delort, 2006; Hu et al., 2007; Hu et al., 2008; Lu et al., 2009), opinionated text (Kim and Zhai, 2009; Ganesan et al., 2010; Paul et al., 2010), or tweets (Yang et al., 2011; Gao et al., 2012; Wei and Gao, 2014; Wei and Gao, 2015; Nguyen and Nguyen, 2016). As far as we know, (Amitay and Paris, 2000) were the first researchers who picked sentences from hyperlinks of a Web document as the summarization. The authors built an InCommonSense system containing hypertext retrieval and description selection (using classification). However the summarization was short because one-sentence from linked texts was selected 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 proposed two context summarization algorithms based on similarity measurements. The first method combined both the content and context of a document and the second only took the context. However, similar with (Amitay and Paris, 2000), the summarization was extracted from the context segments; therefore, it may not completely capture the content of a Web document compared to inherent sentences.

(Sun et al., 2005) addressed the problems of (Amitay and Paris, 2000; Delort et al., 2003) by proposing a system which used the help from click-through data retrieved from search engines to extract important sentences in a Web document. This study based on an assumption that query keywords from users typed on search engines usually reflect the content of a Web document. From this, the authors proposed two methods using an adaptation of significant word

(Luhn, 1958) and latent semantic analysis (Gong and Liu, 2011). This method, however, faces two challenging issues: (i) there is no links from a new Web page to the older ones and (ii) pages which Web users click on are not relevant with their interest.

User-generated contents such as comments were also used to support sentences for generating summarization. (Delort, 2006) clustered comments by using feature vectors and selected summary sentences based on the link of vectors with the clusters. However, this method requires the involvement of human experts to determine the relevance of each cluster. (Hu et al., 2007; Hu et al., 2008) extracted representative sentences in a blog post that best represent the topics discussed among its comments. The authors first derived important words denoted in three graphs: topic, quotation, and mention from comments. Summary sentences were next generated by calculating the distance from each sentence to the graphs. This method, however, only picks up sentences in a blog post as the summarization while ignoring important information from comments. (Lu et al., 2009) studied rated aspect summarization of short comments to help users for better understanding the comments of a target entity. The authors proposed a model containing three steps: aspect discovery and clustering, aspect rating prediction, and representative phrase extraction. Since this method is used to generate the summarization of a target entity, how to adapt it for Web document summarization is still an open question.

Opinionated texts were also investigated and integrated into the summary process. (Kim and Zhai, 2009) studied contrastive opinion summarization in which positively and negatively opinionated sentences were generated from an existing opinion summarizer. The authors formulated the summarization as an optimization problem and proposed two methods that relied on measuring content and contrastive similarity of two sentences. (Ganesan et al., 2010) proposed a framework named Opinosis which used a graph-based approach for abstractive opinionated text summarization. The summarization was generated by scoring various sub-paths in the graph. In the meantime, (Paul et al., 2010) summarized contrastive viewpoints in opinionated text by proposing a two-stages multiple viewpoint model by using an unsupervised probabilistic method. Sentence pairs from opposite viewpoint were scored by using Comparative LexRank algorithm. However, adapting this method to 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 used Support Vector Machines (SVM) and Conditional Random Fields (CRF) as preliminary steps for incorporating tweets into the summarization. The summarization containing both sentences and tweets was generated by a ranking method which approximates an objective function. 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 included a cross-collection topic-aspect modeling (cc-TAM). The cc-TAM was used as a preliminary step to generate a bipartite graph used by co-ranking to select sentences and tweets for multi-document summarization. However, human knowledge of the summarization (features) was not be considered. (Wei and Gao, 2014) integrated the human knowledge of the summarization by proposing 35 features used for a learning to rank model in a news highlight extraction task. The features were defined in three groups: local sentence, local tweet, and cross features. The summarization was generated by selecting top m sentences and tweets after ranking. However, the salient score of a sentence or tweet was computed with the highlights, therefore, it may 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 used auxiliary tweets for building a heterogenous graph random walk (HGRW) to summarize single documents. This method, however, may be sensitive to the noise of data (mentioned by (Erkan and Radev, 2004)) because it bases on LexRank algorithm.

The previous methods exist three issues: (i) supervised approaches need annotated data which is not always available in social context summarization, (ii) unsupervised methods, e.g HGRW are sensitive with data, and (iii) several methods only select sentences or social messages as the summarization. Our method addresses the three issues, in which, firstly, we propose an unsupervised method which treats domain specific and the lack of high-quality annotated data problem in social context summarization. Secondly, we consider the sensitiveness of HGRW by proposing new features which capture textual entailment aspect of a sentence-tweet pair. Finally, the summarization in our method contains both sentences and tweets instead of only selecting sentences. The selected tweets help to enrich information which may not be available in sentences.

3. Summarization by Intra-relation and Inter-relation

This section shows our proposal to select important sentences and representative 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 proposed method for achieving the objective, and show evaluation metric used to compare SoRTESum with state-of-the-art baselines.

3.1. Basic Idea

The data observation and literature review suggested four hypotheses:

- *Representation*: important sentences in a Web document contain important information.
- *Reflection*: representative tweets or comments written by readers reflect document content as well as important 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 some common topics represented in the form of common words.

A Web document (called document) contains a set of sentences in which summary sentences contain important information. The important information of a sentence, s_i, can be measured by a similarity score, e.g. Cosine with the remaining sentences, in which an important sentence receives a higher similarity score compared to unimportant ones. The content of an important sentence is usually mentioned in many tweets indicating that this sentence also receives a lot 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 captures the similarity of a sentence with the remaining ones in the same document and the inter-relation integrates social information.

Inspired by our idea, a summarization framework named SoRTESum was proposed. The framework contains a dual wing entailment graph (DWEG) for modeling sentence-tweet relation denoted by recognizing textual entailment (RTE) (Dagan et al., 2010; Nguyen et al., 2015a). In Figure 2, s_i and t_j denote a sentence and tweet; red lines are inter-relation and blue lines are intra-relation; the weight of each node, e.g. 3.25 at s_1 is a score calculated by intra-relation and inter-relation indicating the importance of a sentence or tweet. In our view, tweets of a document were considered as social information when computing the score of a sentence. Similarity, sentences were also deemed as social information when calculating the score of a tweet. After scoring and ranking, top m ranked sentences and tweets having the highest scores were selected as the summarization.

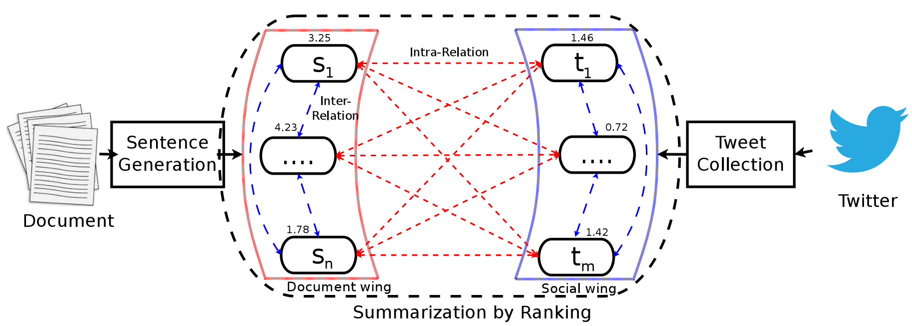


Figure 2: The overview of summarization using intra-relation and inter-relation

Our study is different to (Yang et al., 2011): (i) our method is unsupervised (ranking) instead of 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 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 with traditional methods (Luhn, 1958; Edmundson, 1969; Kupiec et al., 1995; Osborne, 2002; Yeh et al., 2005; Shen et al., 2007): (i) integrating social information and (ii) selecting both important sentences and representative tweets as the summarization instead of only picking up sentences.

3.2. Dataset

DUC (2001, 2002 and 2004)[7] is a well-known data for single and multiple document summarization; this dataset, however, lacks social information. Therefore, we prepared two datasets: (i) a new highlight extraction dataset derived from (Wei and Gao, 2014) and (ii) our dataset collected from Yahoo News.

3.2.1. USAToday and CNN Dataset

A new highlight extraction dataset[8] was derived 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 and CNN. The statistics of this dataset is shown in Table 1. Note that we observed to generate results of two last rows.

3.2.2. Yahoo News Dataset

Since USAToday and CNN dataset has no labels, training a supervised method, e.g. Support Vector Machines (SVM) is challenging. Therefore, a new dataset was created by crawling up-to-date news articles from Yahoo News in May 2015. The dataset contains 157 open-domain news articles along with 3,462 sentences, 5,858 extracted sentences as summaries and 25,633 comments.

[7] http://duc.nist.gov/data.html

[8] http://www1.se.cuhk.edu.hk/∼zywei/data/hilightextraction.zip

Table 1: Statistical observation was taken from (Wei and Gao, 2014); s: sentence and t: tweet.

	Documents	**Highlights**	**Tweets**
# **Total**	121	455	78,419
# Sentence per news	53.6 ± 25.6	3.7 ± 0.4	648 ± 1161.7
# Token per news	1123.0 ± 495.8	49.6 ± 10.0	10364.5 ± 24749.2
# 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 (no stopwords)	s/t: 15.61	—	t/s: 12.62

Two annotators were asked to annotate this dataset in two rounds. In the first round, each annotator read a complete article and selected important sentences. After that, the annotator also read all comments and picked up representative comments. Important sentences and representative comments (called instances) are sentences which mainly reflect the content of a Web document. A selected instance would become a standard summary if the two annotators agree yes; otherwise, it is unimportant. The number of instances is no less than six for documents and 15 for comments. Maximal selected sentences (combining both sentences and comments) are less than 35 for each document.

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

Documents	**Sentences**	**Summaries**	**Comments**
157	3,462	5,858	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	2,393.85
# 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

In the second round, the annotated data was cross-checked to show inter-annotator agreement between 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 0. Finally, the inter-annotator agreement was computed by dividing the completely matched sentences by total extracted sentences. The inter-annotator agreement was defined by Equation (1).

$$agreement = \frac{\#matched\ sentences}{\#extracted\ sentences} * 100 \tag{1}$$

where: $\#matched\ sentences$ are the number of sentences which 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 the Cohen's Kappa[9] between two annotators. The Kappa agreement is 0.5845. The inter-annotator agreement and Kappa score indicate that the agreement of two annotators is moderate. The annotation was conducted in 75 days.

Data statistics in Tables 1 and 2 show: (i) the number of two last rows indicates that there exists 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 comments, i.e. 22.24% of word overlapping in Table 1 and 31.21% in Table 2.

3.2.3. Data Preparation

Tweets and comments with fewer than five tokens were removed because they are too short for summarization. Near-duplicate tweets (those containing similar content) were also removed by Simpson (Nguyen et al., 2015b) in which similar threshold = 0.25 was empirically chosen by running experiments many times. 5-fold cross validation for USAToday and CNN dataset (the same setting with (Wei and Gao, 2014)) and 10-fold cross validation for Yahoo News dataset

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

were used. We selected $m = 4$ for the first dataset because each document has 3-4 highlights and $m = 6$ due to less than 30% average sentences per document (see Table 2) for the second dataset. Stop words, hashtags, links were removed. Extracted sentences and highlight or selected sentences were also stemmed[10] (Porter, 2011).

3.3. Summarization by SoRTESum

This section describes our process to generate the summarization in three steps: feature extraction, calculation modeling 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 similarity measurement may be inefficient due to the noise of data. Therefore, we proposed a set of RTE features[11] represented in the form of two groups: distance and statistical features to calculate the similarity among sentences and tweets. Our features are shown in Table 3.

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

Distance features	**Statistical features**
Manhattan	LCS between s_i and t_j
Euclidean	Inclusion-exclusion coefficient
Cosine similarity	% words of S in T (p(s_i,t_j))
Word matching coefficient (wmc)	% words of T in S (p(t_j, s_i))
Dice coefficient	Word overlap coefficient (woc)
Jaccard coefficient	*Damerau-Levenshtein*
JaroWinkler distance	*Levenshtein distance*

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

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

Distance Features: capture the distance aspect of a sentence-tweet pair, indicating that an important sentence should be close to a representative tweet rather than meaningless ones. Manhattan, Euclidean, and Cosine similarity were defined in Equations (2), (3) and (4).

$$manhattan(\vec{x},\vec{y}) = \sum_{i=1}^{n} \mid x_i - y_i \mid; \tag{2}$$

$$euclidean(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3}$$

$$cos(\vec{x},\vec{y}) = \frac{\vec{x}.\vec{y}}{\parallel \vec{x} \parallel . \parallel \vec{y} \parallel} \tag{4}$$

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.

Word matching coefficient was represented by Equation (5).

$$wmc(s_i,t_j) = comWord(s_i,t_j) \tag{5}$$

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

Dice and Jaccard distance were defined by Equations (6) and (7).

$$dice = \frac{2. \mid X \cap Y \mid}{\mid X + Y \mid}; \tag{6}$$

$$jaccard = \frac{\mid X \cap Y \mid}{\mid X \cup Y \mid} \tag{7}$$

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

JaroWinkler distance of two texts was defined in Equation (8).

$$d_j(s_i,t_j) = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}(\frac{m}{|s_i|} + \frac{m}{|t_j|} + \frac{m-t}{m}) & otherwise \end{cases} \tag{8}$$

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

Suppose s_i can be represented by a and t_j can be denoted by b, DamerauLevenshtein distance was defined in Equation (9).

$$d_{a,b}(i,j) = \begin{cases} max(i,j) & \text{if } min(i,j) = 0 \\ min \begin{cases} d_{a,b}(i-1,j)+1 \\ d_{a,b}(i,j-1)+1 \\ d_{a,b}(i-1,j-1)+1_{(a_i \# b_j)} \\ d_{a,b}(i-2,j-2)+1 \end{cases} & \text{if } i,j > 1 \text{ and } a_i = b_{j-1} \text{ and } a_{i-1} = b_j \\ min \begin{cases} d_{a,b}(i-1,j)+1 \\ d_{a,b}(i,j-1)+1 \\ d_{a,b}(i-1,j-1)+1_{(a_i \# b_j)} \end{cases} & otherwise \end{cases} \tag{9}$$

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 the respective symbols are the same; $d_{a,b}(i-2,j-2)+1$ corresponds to a transposition between two successive symbols.

Levenshtein distance[12] was defined in Equation (10).

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

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

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 word overlapping between a sentence and a tweet. An important sentence and a representative tweet usually contain common words (the *generation* hypothesis), indicating their content is similar. The longest common substring of two texts was defined in Equation (11).

$$lsc(s_i, t_j) = \frac{len(maxComSub(s_i, t_j))}{min(len(s_i), length(t_j))} \tag{11}$$

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

The inclusion_exclusion coefficient was denoted by Equation (12).

$$\text{inclusion-exclusion}(s_i, t_j) = \frac{comWord(s_i, t_j)}{len(s_i) + len(t_j)} \tag{12}$$

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.

The percentage of word overlapping of s_i in t_j was defined by Equation (13).

$$p(s_i, t_j) = \frac{k}{len(s_i)} \tag{13}$$

where: k is the set 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 was also defined by changing the role of s_i and t_j.

Word overlap coefficient was defined by Equation (14).

$$woc(s_i, t_j) = \frac{wmc(s_i, t_j)}{min(length(s_i), len(t_j))} \tag{14}$$

where: $wmc()$ is the word matching coefficient of two texts s_i and t_j defined in Equation (5), $len()$ returns a set of words of a text.

3.3.2. Calculation Modeling

The proposed features were applied to calculate the score of each sentence and tweet in Figure 2. More precisely, two methods named *SoRTESum Iter-Wing* and *SoRTESum Dual-Wing* were proposed.

SoRTESum Inter-Wing: In this method, the score of a sentence or a tweet was computed by using auxiliary information from the other side. For example, the score of sentence s_i was calculated by using auxiliary tweet t_j on the tweet side. The calculation was defined in Equation (15).

$$score(s_i) = \frac{1}{m}\sum_{j=1}^{m} rteInterScore(s_i, t_j) \tag{15}$$

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. The entailment score was calculated by Equation (16).

$$rteInterScore(s_i, t_j) = \frac{1}{F}\sum_{k=1}^{F} f_k(s_i, t_j) \tag{16}$$

where: F contains 14 RTE features; $f()$ is a similarity function calculated by each k^{th} feature. Similarly, the score of a tweet was also computed in the same mechanism in Equation (17)

$$score(t_j) = \frac{1}{n}\sum_{i=1} rteInterScore(t_j, s_i) \tag{17}$$

where: n is a set of sentences in a document d.

SoRTESum Dual-Wing: In this method, the RTE value of a sentence was calculated 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, this score was also computed by using auxiliary tweets t_1 to t_m. The final RTE score of a sentence was summed through a balanced parameter. The calculation was defined in Equation (18).

$$score(s_i) = \delta * \sum_{k=1}^{n} rteIntraScore(s_i, s_k) + (1-\delta) * \sum_{j=1}^{m} rteInterScore(s_i, t_j) \tag{18}$$

Similarly, the RTE score of a tweet was also computed in the same mechanism in Equation (19).

$$score(t_j) = \delta * \sum_{k=1}^{m} rteIntraScore(t_j, t_k) + (1-\delta) * \sum_{i=1}^{n} rteInterScore(t_j, s_i) \quad (19)$$

where: δ is a balanced parameter which controls the contribution of social information to the summary process; n and m are the number of sentences and tweets. In this view, tweets were used to support our model in finding important sentences and also, sentences were deemed as social information to help our method in selecting representative tweets. Note that $rteIntraScore(s_i, t_j)$ was also computed by Equation (16). Choosing balanced parameter δ is shown in Section 4.3.

3.3.3. Summarization

The summarization was generated by selecting vertices having the highest scores in DWEG. The selection was denoted in Equation (20).

$$S_r \leftarrow ranking(S); \quad T_r \leftarrow ranking(T) \quad (20)$$

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 were selected as the summarization.

3.4. Statistical Analysis

3.4.1. Baseline

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

- **Random Method**: selects sentences and tweet or comments randomly as the summarization.
- **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-based method for computing relative importance of textual units in text summarization. LexRank considers extractive

text summarization relying on the concept of sentence salience to identify the most important sentences in a document, in which the salience was typically defined by terms. In this study, LexRank algorithm[13] was applied with using tokenization and stemming[14].

- **Learning to Rank (L2R)**: was applied by (Wei and Gao, 2014). The authors proposed 35 features and adopted RankBoost implemented in RankLib[15] for training a learning to rank model with ERR metric score and 300 iterations. More precisely, the authors separately trained two learning to rank models, one for sentences and the other for tweets. In training, when modeling a sentence, a set of social features from tweets was combined with local features of this sentence. Similarity, a set of social features from sentences were also used to help local features when modeling a tweet. Unnecessary features, e.g. hashtags, URLs or quality depend were ignored because they are not usually available in comments. This method contains two baselines: L2R only using local features for sentences or tweets/comments (L2R), and L2R using local and cross features (CrossL2R).
- **SVM:** was proposed by (Cortes and Vapnik, 1995) and used by (Yang et al., 2011; Kupiec et al., 1995; Osborne, 2002; Yeh et al., 2005). The authors trained a binary classifier on training data and applied the classifier on testing data to create the summarization. The summarization was generated by selecting sentences or comments labeled by 1. In our study, LibSVM[16] was used with RBF kernel, features were scaled in [-1, 1]; comments were weighted by 85% due to the imbalanced data (see Table 2). Note that this method was only used for Yahoo News dataset because labels were not available in USAToday and CNN dataset.

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

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

[15] https://people.cs.umass.edu/~vdang/ranklib.html

[16] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

- **RTE One Wing:** uses one wing information (document or tweet/comment) to calculate the RTE score. For example, this 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 comment.

3.4.2. Evaluation Metric

In USAToday and CNN dataset, highlights were used as standard summaries. In Yahoo News dataset, selected sentences and comments (those which were labeled by 1 in the annotation step) were used as standard summaries. For evaluation, F-1 ROUGE-N[17] (Lin and Hovy, 2003)(N=1, 2) was employed, in which ROUGE-N was defined in Equation (21) (Lin and Hovy, 2003).

$$ROUGE - N = \frac{\sum_{s \in S_{ref}} \sum_{gram_n \in s} Count_{match}(gram_n)}{\sum_{s \in S_{ref}} \sum_{gram_n \in s} Count(gram_n)} \quad (21)$$

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.

4. Results and Discussion

In order to measure our success, in Section 4.1 we show comparison results of SoRTESum with state-of-the-art baselines. The comparison answers two questions: (i) whether the performance of our approach can compare to other methods and (ii) whether our approach is efficient. Sections 4.2 and 4.3 investigate feature contribution and the role of trade-off parameter in Equations (18) and (19). We also illustrate the position distribution observation of sentences and tweets generated from our method. This observation reveals the role of sentence position in our model. We finally validate our hypotheses and deeply analyze our model by a running example.

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

4.1. Experimental Results

Our method was first evaluated on USAToday and CNN dataset. Results in Table 4 show that SoRTESum clearly outperforms baselines from 0.51% to 8.8% in ROUGE-1 of document summarization, except for CrossL2R. In tweet summarization, our method is the best. This supports our idea and hypotheses stated in Section 3.1. The performance of document summarization is better than that in tweet summarization because tweets were usually generated from document content (a similar with (Yang et al., 2011; Wei and Gao, 2014; Nguyen and Nguyen, 2016)) supporting the *reflection* hypothesis.

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

System	**Document**		**Tweet**	
	F-1 R-1	F-1 R-2	F-1 R-1	F-1 R-2
Random	0.167	0.037	0.156	0.059
Sentence Lead	0.249	0.096	—	—
LexRank	0.183	0.045	0.154	0.056
L2R*	0.248	0.086	0.199	0.064
CrossL2R*	**0.270**	**0.111**	**0.209**	*0.069*
RTE One Wing	0.202	0.072	0.191	0.067
SoRTESum Inter-Wing	*0.255*	*0.098*	*0.201*	0.068
SoRTESum Dual-Wing	*0.254*	0.096	**0.209**	**0.074**

SoRTESum outperforms L2R (Wei and Gao, 2014) in both ROUGE-1, 2 of document and tweet summarization even though L2R is a supervised method. This shows the efficiency of our approach and features. In other words, our method comparably performs with CrossL2R (Wei and Gao, 2014) in both ROUGE-1, 2. This is because: (i) CrossL2R is also a supervised method and (ii) a salient score of a sentence or tweet in CrossL2R was computed by a maximal ROUGE-1 F-score between this instance and corresponding ground-truth highlights. As the result, this model tends to select sentences and tweets which are highly similar with the highlights. However, even with this, in document summarization, our model still obtains a comparable result, i.e. 0.255 vs. 0.270 in ROUGE-1 and 0.098 vs. 0.111 in ROUGE-2. In tweet summarization, SoRTESum obtains the same result, i.e. 0.209 in ROUGE-1 but in ROUGE-2, conversely, SoRTESum is the best (0.074 vs. 0.069). This shows that our approach is appropriate for tweet summarization and supports our hypotheses stated in Section 3.1.

The performance of SoRTESum Inter-Wing is the same with SoRTESum Dual-Wing in document summarization, i.e. 0.255 vs. 0.254 in ROUGE-1; in tweet summarization, however, SoRTESum Dual-Wing dominates SoRTESum Inter-Wing (0.209 vs. 0.201). This is because the score of a tweet in SoRTESum Dual-Wing was calculated by accumulating the scores from corresponding sentences and remaining tweets. As the result, the model tends to select longer tweets. Note that, the performance is slightly different.

SoRTESum was extensively validated on our dataset. In Table 5, our method is competitive with CrossL2R in ROUGE-1 of document summarization (0.362 vs. 0.363). This shows that the performance of our method can reach to supervised methods. In comment summarization, interestingly, LexRank is the strongest method. This is because: (i) comments are more formal than tweets (see Table 4) and (ii) the number of comments is large enough for LexRank algorithm (163 comments per document, see Table 2). However, LexRank is sensitive with data (Erkan and Radev, 2004) and obtains quite poor results in Table 4 supporting this conclusion. Our methods obtain competitive results in both ROUGE-1 and ROUGE-2 of document and comment summarization compared to LexRank.

Results from Tables 4 and 5 indicate that SoRTESum outperforms L2R and SVM, two supervised methods. This proves the efficiency of our method and features. SoRTESum dominates LexRank in almost cases due to the integration of social information (the same conclusion with (Nguyen and Nguyen, 2016; Wei and Gao, 2015; Wei and Gao, 2014)). SentenceLead is a competitive baseline because it simulates the summarization by picking up some first sentences

Table 5: Summary performance on Yahoo News Dataset

System	Document		Comment	
	F-1 R-1	F-1 R-2	F-1 R-1	F-1 R-2
Random	0.272	0.201	0.103	0.045
Sentence Lead	0.360	*0.309*	—	—
LexRank	0.328	0.257	**0.244**	**0.140**
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
RTE One Wing	0.352	0.294	0.222	0.118
SoRTESum Inter-Wing	0.357	0.299	*0.237*	*0.135*
SoRTESum Dual-Wing	*0.362*	0.302	0.206	0.113

(Nenkova, 2005). SVM achieves quite poor results due to the noise of mixing features as similar as L2R. RTE-Sum One Wing obtains competitive results because it uses our proposed features. Experimental results also validate our hypotheses: representation, reflection, generation and common topics, in which we use these hypotheses to formulate and calculate the score of sentence and tweet or comment.

We discuss important different points of our method with (Wei and Gao, 2015), which uses heterogenous graph random walk (HGRW) (using ranking and the same dataset) due to experimental settings and re-running experiments. Firstly, HGRW is a variation of LexRank, which utilizes *IDF-modified-cosine similarity*; therefore, the noise of data may negatively affect the summarization (the same conclusion with (Erkan and Radev, 2004)). The results of LexRank in Tables 4 and 5 and the performance of CrossL2R-T and HGRW-T (decreasing from 0.295 to 0.293, see (Wei and Gao, 2015)) support this conclusion. On the other hand, our method combines a set of features helping to avoid the noise of data; hence, the performance increases from 0.201 to 0.209 in ROUGE-1 of tweet summarization (Table 4). *IDF-modified-cosine similarity* needs a large corpus to calculate TF and IDF using bag-of-words model (Erkan and Radev, 2004) whereas our approach only requires a single document and its social information to extract important sentences and tweets. This shows that our method is insensitive to the number of documents as well as tweets. Results in Table 4 support our conclusion. In addition, new features, e.g. word2vec similarity can be easy integrated into our model while adding new features into the *IDF-modified-cosine similarity* is still an open question. Finally, their work considers the impact of tweet volume and latency for sentence extraction. It is difficult to take these values for news or forum comments. In this sense, our method can be flexible to adapt for unrestricted domains.

All models in Tables 4 and 5 are inefficient with informal social messages, i.e. very short, abbreviated, or ungrammatical tweets or comments. This can be possibly solved by integrating a sophisticated pre-processing step. In addition, tweets or comments did not come from the news sources challenge all methods because there is no content consistency between sentences and social messages. This can be solved by integrating a sophisticated crawling method to capture relevant information from other sources. Our method is also limited if the content of sentence and tweets or comments is highly abstractive, e.g. need inference. In this case, a more novel approach, e.g. RTE should be considered.

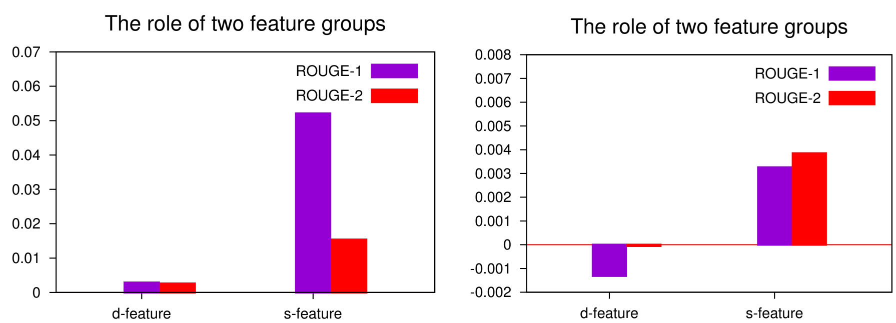
4.2. Feature Contribution Analysis

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 performance minus of SoRTESum Dual-Wing using all features with the model using $n-1$ features. Top five effective features are shown.

Table 6 indicates that both distance and statistical features affect the summarization. In document summarization, statistical features (in italic) play an important role. This shows that important sentences include important common words or phrases. In document and tweet summarization, Dice coefficient, Inclusion-exclusion coefficient, and Jaccard positively affect the summarization. In tweet summarization, however, distance features are more important than the remaining ones (only Inclusion-exclusion coefficient appearing).

Table 6: Top five effective features generated from SoRTESum Dual-Wing; * is Inclusion-exclusion coefficient; *italic* denotes statistical features.

Feature	**Document**		**Feature**	**Tweet**	
	ROUGE-1	ROUGE-2		ROUGE-1	ROUGE-2
Overlap	$0.23 \text{ x } 10^{-2}$	$0.25 \text{ x } 10^{-2}$	Euclidean	$0.4 \text{ x } 10^{-3}$	$0.3 \text{ x } 10^{-3}$
Dice	$0.21 \text{ x } 10^{-2}$	$0.14 \text{ x } 10^{-2}$	Dice	$0.38 \text{ x } 10^{-4}$	$0.203\text{x}10^{-4}$
*In-ex coeffi**	$0.7 \text{ x } 10^{-3}$	$0.7 \text{ x } 10^{-3}$	*In-ex coeffi**	$0.33\text{x}10^{-4}$	$0.203\text{x}10^{-4}$
Jaccard	$0.4 \text{ x } 10^{-3}$	$0.5 \text{ x } 10^{-3}$	Jaccard	$0.33\text{x}10^{-4}$	$0.203\text{x}10^{-}4$
Matching	$0.1 \text{ x } 10^{-3}$	$0.3 \text{ x } 10^{-3}$	Manhattan	$0.1\text{x}10^{-5}$	$0.4\text{x}10^{-5}$



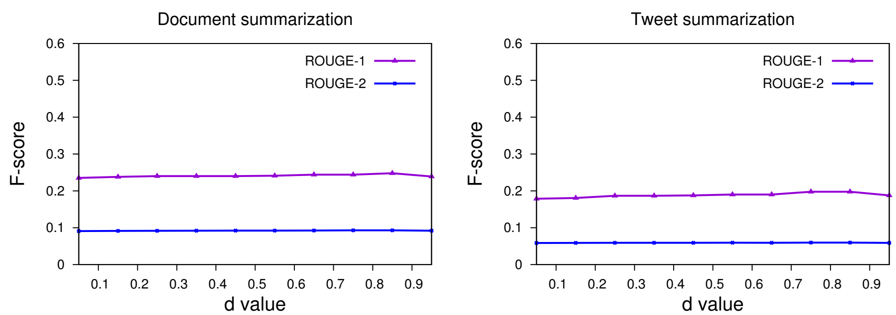
(a) Feature group in document (b) Feature group in tweet

Figure 3: The contribution of feature groups in SoRTESum Dual-Wing

The contribution of distance (d-feature) and statistical feature (s-feature) group was also investigated. The F-score ratio (y-axes) was computed by the performance minus of SoRTESum Dual-Wing using two feature groups with the model using one group. In Figure 3, statistical features positively affect document and tweet summarization, i.e. 0.05 and 0.004, whereas distance feature has a negative influence in tweet summarization in Figure 3b. This concludes that statistical features play an important role in document summarization. Although each distance feature in Table 6 has a positive impact, combining them may lead to feature conflict.

4.3. Tuning Trade-off Parameter

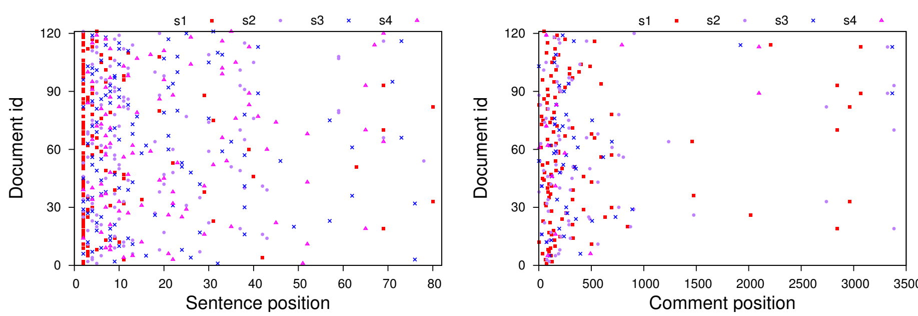
The impact of balanced parameter in Equations (18) and (19) was investigated by adjusting δ in [0.05..0.95] with jumping step = 0.1. Results from Figure 4 show that when δ increases, auxiliary information benefits the performance of our model until some turning points. The performance generally improves when δ closes to 0.85. After that, the performance slightly drops because when $\delta > 85$, the model is nearly the same with SoRTESum Inter-Wing. We, therefore, empirically selected $\delta = 0.85$. Note that the change is not much different among tuning points because the score of a sentence or tweet was computed by averaging RTE features; therefore, the role of δ may be saturated.



(a) Performance on document side (b) Performance on tweet side

Figure 4: Parameter adjustment of δ of SoRTESum Dual-Wing

4.4. Sentence Position Observation



(a) Document summarization (b) Tweet summarization

Figure 5: The position of summary sentences and tweets

We further investigated the position of extracted sentences from SoRTESum Dual-Wing. From Figure 5, we observe that most important sentences are located within 15 first sentences on the document side and 300 on the tweet side. There are also some outlier points, e.g. 80^{th} in Figure 5a and 3.500^{th} in Figure 5b because several documents contain a larger number of sentences and tweets (thousand tweets). Considering data observation in Tables 1 and 2, we conclude that: (i) the density distribution of tweets is scattered because the sequence aspect does not explicitly exist on the tweet side and (ii) SentenceLead is inefficient in tweet or comment summarization because representative social messages usually appear in a wider range compared to sentences.

4.5. Hypothesis Analysis

Our hypotheses in Section 3.1 were deeply analyzed 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.

Table 7: An example of Boston Bombing (24) in USAToday and 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

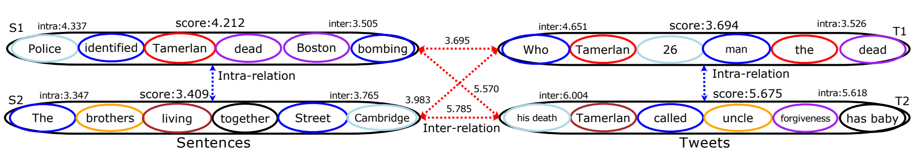


Figure 6: A running example from Table 2 implemented from SoRTESum Dual-Wing.

Figure 6 indicates that important sentences, i.e. S_1 and T_2 receive a higher score whereas non-summary sentences, i.e. S_2 and T_1 obtain a lower score.

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

4.6. Error Analysis

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.	[-] T1: I proudly say I was the 1st 1 to write this on twitter. Uncle,Tamerlan Tsarnaev called, asked for forgiveness.
[-] T2: 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: 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: Tamerlan Tsarnaev ID'd as dead Boston blast suspect - USA Today - USA TODAY, Tamerlan Tsarnaev ID'd as dead.	[+] T4: #BostonMarathon bomber Tamerlan called uncle couple of hours before he was shot dead said 'I love you and forgive me.

An error analysis was also deeply conducted to show the limitation of our method. In Table 8 (the Web interface can be seen at SoRTESum system[18]), both the two methods yield the same results in document summarization, in which *S1*, *S2*, and *S3* are summary sentences. Clearly, 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 tweet summarization, two methods generate three the same tweets and the remaining one is different. The summarization contains the same tweet, i.e. *T1* in SoRTESum Inter-Wing and *T3* in SoRTESum Dual- Wing; the other ones are different making the difference of summarization performance between two methods. They are quite relevant to this event but do not directly mention the death of Tamerlan Tsarnaev, e.g. *T2*. This is because $T2$ also include important information; hence this challenges our method and leads to the lower performance.

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

5. Conclusion

This paper presents *SoRTESum*, a novel ranking summary framework which utilizes the social information of a Web document to generate high-quality summarization. Our framework combines intra-relation and inter-relation to calculate the score of each sentence or tweet and ranks to select top m sentences and

[18]http://150.65.242.101:9293

tweets as the summarization. This paper concludes that integrating social information and formulating a sentence-tweet pair by a set of features benefit the summarization. In the first aspect, social information supports local information to improve the quality of the summary process. The social information not only come from tweets or comments but also from sentences due to mutual reinforcement support. In the second aspect, combining features help to avoid the noise of data which may appear when using one feature, e.g. Cosine. Our method is extensively evaluated on two datasets: a new 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 direction, 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 semantics aspect. To ensure the quality of the summarization, human evaluation should also be considered.

Acknowledgment

This work was supported by JSPS KAKENHI Grant number 3050941, JSPS KAKENHI Grant Number JP15K12094. 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 anonymous reviewers for their detailed comments 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.

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), 105–105.

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. (2011). 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.

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

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.

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

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., Ha, Q. T., Nguyen, T. D., Nguyen, T. T., & Nguyen, L. M. (2015a). 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. (2015b). 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.

Osborne, M. (2002). Using maximum entropy for sentence extraction. In *Proceedings of the ACL-02 Workshop on Automatic Summarization* (Volume 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. In *IJCAI*, (Vol.7, pp. 2862–2867).

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

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

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

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.