SoRTESum: A Social Context Framework for Single-Document Summarization

Minh-Tien Nguyen[1,2] and Minh-Le Nguyen[1]

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

Abstract. The combination of web document contents, sentences and users' comments from social networks provides a viewpoint of a web document towards a special event. This paper proposes a framework named *SoRTESum* to take advantage of information from Twitter viz. diversity and reflection of document content to generate high-quality summaries by a novel sentence similarity measurement. The framework first formulates sentences and tweets by recognizing textual entailment (RTE) relation to incorporate social information. Next, they are modeled in a Dual Wing Entailment Graph, which captures the entailment relation to calculate the sentence similarity based on mutual reinforcement information. Finally, important sentences and representative tweets are selected by a ranking algorithm. By incorporating social information, *SoRTESum* obtained improvements over state-of-the-art unsupervised baselines e.g., Random, SentenceLead, LexRank of 0.51% - 8.8% of ROUGE-1 and comparable results with strong supervised methods e.g., L2R and CrossL2R trained by RankBoost for single-document summarization.

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

1 Introduction

Thanks to the growth of social networks e.g., Twitter[3], users can freely express their opinions on many topics in the form of tweets - short messages, maximum 140 letters. For example, after reading a web document which mentions a special event, e.g., Boston bombing, readers can write their tweets about the event on their timeline. These tweets, called social information [18] not only reveal reader's opinions but reflect the content of the document and describe facts about the event. From this observation, an interesting idea is that social information can be utilized as mutual reinforcement for web document summarization.

Given a web document, the summarization has to extract important sentences [10] by using statistical or linguistic information. Existing methods, however, only consider inherent document information as sentence or word/phrase

[3] http://twitter.com - a microblogging system

level while ignoring the social information. How to elegantly formulate sentence-tweet relation and how to effectively generate high-quality summaries using social information are challenging questions.

Social context summarization can be solved by several approaches: topic modeling [3,9]; clustering [15,6]; graphical model [5,4]; or ranking [7,17]. Yang et al. proposed a dual wing factor graph model (DWFG) for incorporating tweets into the summarization [18]. The author used classification as a preliminary step in calculating weight of edges for building the graph. Wei et al. used ranking approach with 35 features trained by RankBoost for news highlight extraction [16]. However, lack of high-quality annotated data challenges supervised learning methods [18,16] to solve the summariation. In contrast, Wei et al. proposed a variation of LexRank, which used auxiliary tweet information in a heterogenous graph random walk (HGRW) to sumarize single documents [17].

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

- We propose to formally define sentence-tweet relation by Recognizing Textual Entailment (RTE). The relation is different in comparison to sentence-tweet representation in [18,16,17]. To the best of our knowledge, no existing methods solve social context summarization by RTE.
- We conduct a careful investigation to extract 14 features of RTE in the form of two groups: distance and statistical features. This provides a feature selection overview for the summarization using RTE.
- We propose a unified framework which utilizes RTE features for calculating sentence/tweet similarity. The framework is compared to several baselines and promising results indicate that our method can be successfully applied for summarizing web documents.

To solve the social context summarization, three hypothesis are considered: (1) *representation*: important sentences in a web document represent its content; (2) *reflection*: representative tweets written by readers reflect document content as well as important sentences and (3) *generation*: readers tend to use words/phrases appearing in a document to create their comments. Given a web document and tweets generated by readers after reading the document, the framework first calculates similarity score of each sentence by using RTE features with additional social information from tweets. Next, similarity score of each tweet is computed in the same mechanism using additional information from sentences. Finally, important sentences and representative tweets having the highest score are selected by a ranking algorithm as summaries.

The rest of this paper is organized as follows: Section 2 will show our approach to satisfy the goal along with idea, feature extraction and model; Section 3 will illustrate experimental results, and give discussion along with error analysis; the final section is conclusion.

2 Summarization by Ranking

This section shows our proposal of social context summarization by ranking in three steps: basic idea, feature selection and summarization.

2.1 Basic Idea

Cosine similarity can be used to incorporate document-social information, however, the noise of tweets e.g., hashtags, emoticons can badly affect the similarity calculation. We therefore propose to utilize RTE for representing sentence-tweet relation with rich features. Given a text T and hypothesis H, RTE is a task of deciding whether the meaning of H can be plausibly inferred from T in the same context and denoted by an unidirectional relation as $T \rightarrow H$ [1]. We extend RTE from unidirectional to bidirectional relation as $T \leftrightarrow H$, in which T is a sentence s_i and H is a tweet t_j. The "$\leftrightarrow$" means existing a content similarity[4] between s_i and t_j.

We define a variation of Dual Wing Entailment Graph (DWEG) from [18] for modeling sentence-tweet relation in a social context. In the DWEG, vertices are sentences and tweets; edges are the sentence-tweet relation; and weight is the RTE similarity value. Given a DWEG G_d, our goal is to select important sentences and the most representative tweets which mainly reflect the content of a document.

Our study has important differences in comparison to [18]. Firstly, our method is unsupervised (ranking) instead of classification. Secondly, we use RTE instead of three types of sentence-tweet relation. Our approach is similar to the method of Wei et al. [16,17] (using ranking) as well as the dataset. However, representing sentence-tweet by RTE and ranking to generate summaries are two key differences in comparison to [16], which used RankBoost, another supervised method. In addition, our method calculates inter-wing/dual-wing similarity with a set of RTE features instead of IDF-modified-cosine similarity in comparison to [17].

2.2 Feature Extraction

To detect the entailment, a straightforward method is to define a set of features for representing sentence-tweet relation. Term frequency - inverse document frequency (TF-IDF) or Cosine similarity can be used; however, they may be inefficient for the summarization due to the noise of data. As the main contribution, we proposed to use a set of features derived from [12] in the form of two groups: distance and statistical features shown in Table 1.

Distance Features: These 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.

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

Table 1: The features; italic in the second column is the distance features; S is a sentence, T is a tweet; LCS is the longest common sub string

Distance Features	**Statistical Features**
Manhattan	LCS between S and T
Euclidean	Inclusion-exclusion coefficient
Cosine similarity	% words of S in T
Word matching coefficient	% words of T in S
Dice coefficient	Word overlap coefficient
Jaccard coefficient	*Damerau-Levenshtein*
Jaro coefficient	*Levenshtein distance based on word*

Statistical Features: 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 it has similar content.

2.3 Summarization

The goal of our approach is to select important sentences and representative tweets as summaries because they provide more information regarding the content of a document rather than only providing sentences. In our method, tweets are utilized to enrich the summary when calculating the score of a sentence, and sentences are also considered as mutual reinforcement information in computing the score of a tweet. More precisely, the weight of each instance is computed by the entailment relation using the features and the top K instances, which have the highest score will be selected as the summaries.

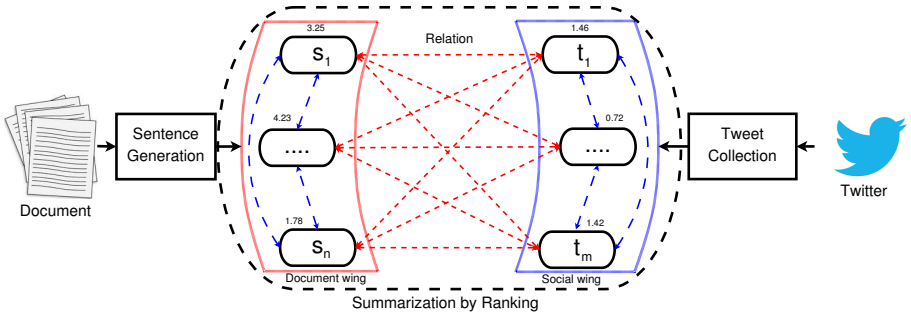


Fig. 1: The overview of summarization using DWEG; s_i and t_j denote a sentence and a tweet in document and social wing; red lines are inter-relation and blue lines are intra-relation; weight of each node (e.g., 3.25 at s_1) is the entailment value.

The framework in Figure 1 calculates entailment weight by an iterative algorithm as an accumulative mechanism to decide whether a sentence is important or not. More precisely, we proposed two methods named *RTE-Sum inter wing* and *RTE-Sum dual wing*.

RTE-Sum Inter Wing: In this method, a sentence weight was computed by using additional information from tweets. For example, the weight of s_i was calculated by relevant tweet t_j on the tweet side. The score of t_j was also computed as the same mechanism. The calculation is shown in Eq. (1).

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

where: $s_i \in S$, $t_j \in T$; $rteScore(s_i, t_j)$ returns the entailment value between sentence s_i and tweet t_j; m is the number of sentences/tweets corresponding to each document. The entailment score is calculated by Eq. (2).

$$rteScore(s_i, t_j) = \frac{1}{F} \sum_{k=1}^{F} f_k(s_i, t_j) \tag{2}$$

where: F is the number of features; f is the similarity function calculated by each feature.

RTE-Sum Dual Wing: In this method, a sentence RTE score was calculated by using remaining sentences as the main part (intra-relation) following tweets as auxiliary information (inter-relation) in an accumulative mechanism. For example, the score of s_i was calculated by s_1 to s_n; at the same time, the score was also computed by relevant tweets t_1 to t_m. Finally, RTE value of a sentence was average of all entailment values. The calculation is shown in Eq. (3).

$$score(s_i) = \delta * \sum_{k=1}^{n} rteScore(s_i, s_k) + (1 - \delta) * \sum_{j=1}^{m} rteScore(s_i, t_j) \tag{3}$$

RTE value of a tweet was also computed as the same mechanism in Eq. (4).

$$score(t_j) = \delta * \sum_{k=1}^{m} rteScore(t_j, t_k) + (1 - \delta) * \sum_{i=1}^{n} rteScore(t_j, s_i) \tag{4}$$

where δ is the damping factor; n and m are the number of sentences and tweets.

Ranking: Important sentences and representative tweets were found by selecting the highest score of vertices in the DWEG. The selection is denoted in Eq. (5).

$$S_r \leftarrow ranking(S_n);\ T_r \leftarrow ranking(T_m) \tag{5}$$

where *ranking()* returns a list of instances in a decreased weight order; *top-K* instances would be selected as the summaries from the S_r and T_r.

3 Results and Discussion

3.1 Experimental Setup

The dataset in [16][5] was used for evaluation. It contains 121 documents with 455 highlights and 78.419 tweets in 17 salient news events taken from CNN[6] and USAToday[7]. Each article includes three or four highlights which were manually selected by human. The detail can be seen in [16].

Comments less than five tokens were removed. Near-duplicate tweets (those containing similar content) were also removed by Simpson [13]; similar threshold 0.25 was obtained by running our experiments many times. The damping factor δ will be shown in Section 3.6. 5-fold validation with $K = 4$ is conducted in evaluation; stop words, hashtags, links were removed; and summary instances were also stemmed[8] [14].

3.2 Baselines

The following systems were used to compare to SoRTESum:

- **Random Method**: selects sentences and comments randomly.
- **SentenceLead:** chooses the first x sentences as the summarization [11].
- **LexRank:** summarizes a given news article using LexRank algorithm[9] [2].
- **L2R**: uses RankBoost with local and cross features [16], using RankLib[10].
- **Interwing-sent2vec:** uses Cosine similarity; by Eq. (1). A sentence to vector tool was utilized to generated vectors[11] ($size = 100$ and $window = 5$) with 10 million sentences from Wikipedia[12].
- **Dualwing-sent2vec:** uses Cosine similarity; by Eq. (3) and (4).
- **RTE One Wing:** uses one wing (document/tweet) to calculate RTE score.

3.3 Evaluation Method

Highlight sentences were used as standard summarization of evaluation by using ROUGE-N[13] (N=1, 2) [8] with stemming and removing stopwords.

3.4 Results and Discussion

Results in Tables 2 and 3 show that our method clearly outperforms the baselines by 0.51%-8.8% in the document side in ROUGE-1, except for CrossL2R. In

[5] http://www1.se.cuhk.edu.hk/∼zywei/data/hilightextraction.zip
[6] http://edition.cnn.com
[7] http://www.usatoday.com
[8] http://snowball.tartarus.org/algorithms/porter/stemmer.html
[9] https://pypi.python.org/pypi/sumy/0.3.0
[10] https://people.cs.umass.edu/∼vdang/ranklib.html
[11] https://github.com/klb3713/sentence2vec/blob/master/demo.py
[12] https://meta.wikimedia.org/wiki/Data_dump_torrents
[13] http://kavita-ganesan.com/content/rouge-2.0-documentation

Table 2: Document summarization; * is supervised methods; bold is the best value; italic is compared value to the best.

	ROUGE-1			**ROUGE-2**		
System	**Avg-P**	**Avg-R**	**Avg-F**	**Avg-P**	**Avg-R**	**Avg-F**
Random	0.140	0.205	0.167	0.031	0.044	0.037
Sentence Lead	0.196	0.341	0.249	0.075	0.136	0.096
LexRank	0.127	0.333	0.183	0.030	0.088	0.045
Interwing-sent2vec	0.208	0.315	0.250	0.069	0.116	0.086
Dualwing-sent2vec	0.148	0.194	0.168	0.044	0.058	0.050
RTE-Sum one wing	0.137	0.385	0.202	0.048	0.143	0.072
L2R* [16]	0.202	0.320	0.248	0.067	0.120	0.086
CrossL2R* [16]	0.215	0.366	**0.270**	0.086	0.158	**0.111**
SoRTESum inter wing	0.189	0.389	*0.255*	0.071	0.158	*0.098*
SoRTESum dual wing	0.186	0.400	*0.254*	0.068	0.162	0.096

Table 3: Tweet summarization, SentenceLead was not used

	ROUGE-1			**ROUGE-2**		
System	**Avg-P**	**Avg-R**	**Avg-F**	**Avg-P**	**Avg-R**	**Avg-F**
Random	0.138	0.179	0.156	0.049	0.072	0.059
LexRank	0.100	0.336	0.154	0.035	0.131	0.056
Interwing-sent2vec	0.177	0.222	0.197	0.055	0.071	0.062
Dualwing-sent2vec	0.153	0.195	0.171	0.039	0.055	0.046
RTE-Sum one wing	0.145	0.277	0.191	0.054	0.089	0.067
L2R* [16]	0.155	0.276	0.199	0.049	0.089	0.064
CrossL2R* [16]	0.165	0.287	**0.209**	0.053	0.099	*0.069*
SoRTESum inter wing	0.154	0.289	*0.201*	0.051	0.104	*0.068*
SoRTESum dual wing	0.161	0.296	**0.209**	0.056	0.111	**0.074**

addition, the performance in the document side is better than that on the tweet side because comments are usually generated from document content (similarly [18]) supporting the *Reflection* hypothesis stated in Section 1.

SoRTESum outperforms L2R [16] in both ROUGE-1, 2, on document and tweet side even though it is a supervised method. This shows the efficiency of our approach as well as the features. In other words, our method performs comparably CrossL2R [16] in both ROUGE-1, 2. This is because (1) CrossL2R is also a supervised method; and (2) a salient score of an instance in [16] was computed by maximal ROUGE-1 between this instance and corresponding ground-truth highlight sentences. As the results, this model tends to select highly similar sentences and tweets with the highlights improving the overall performance of the model. However, even with this, our models still obtain comparable result of 0.255 vs. 0.270 in document side and the same result of 0.209 on tweet side of ROUGE-1. In ROUGE-2, although CrossL2R slightly dominates SoRTESum on document side (0.111 vs. 0.098), on tweet side, conversely, SoRTESum outper-

forms 0.05% (0.074 vs. 0.069). This shows that out approach is also appropriate for tweet summarization and supports our hypothesis stated in Section 1.

We discuss some important different points with [17] (using same method - ranking and the dataset) due to experimental settings and re-running the experiments. Firstly, [17] uses *IDF-modified-cosine similarity*, thus the noise of tweets may badly affect to the summarization (same conclusion with [2]). The performance of CrossL2R-T and HGRW-T supports this conclusion (decreasing from 0.295 to 0.293, see [17]). On the other hand, our method combines a set of RTE features helping to avoid the tweet noise; hence, the performance increases from 0.201 to 0.209. In addition, the *IDF-modified-cosine similarity* needs a large corpus to calculate TF and IDF with a bag of words [2] whereas our approach only requires a single document and its social information to extract important sentences. This shows that our method is insensitive to the number of documents as well as tweets. 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 tweets volume and latency for sentence extraction. It is difficult to take these values for news comments as well as forum comments. In this sense, our method can be flexibly to adapt for other domains. Of course, tweets did not come from the news sources challenge both the two methods because there is no content consistency between sentences and tweets. However, we guess that even in this case, our method may be still effective because SoRTESum captures words/tokens overlapping based on a set of features while only using *IDF-modified-cosine similarity* may limit HGRW. In another words, both the two methods are inefficient in dealing informal tweets e.g., very short, abbreviated, or ungrammatical tweets. It is possibly solved by integrating a sophisticated preprocessing step.

The performance of *SoRTESum inter wing* is the same with *SoRTESum dual wing* on document side of ROUGE-1 (0.255 vs. 0.254); on tweet side, however, *SoRTESum dual wing* dominates *SoRTESum inter wing* (0.209 vs. 0.201). This is because a score of a tweet in *SoRTESum dual wing* was calculated by accumulating corresponding sentences and remaining tweets; therefore, long instances obtain higher scores. As a result, the model tends to select longer instances. However, the performance is not much different.

SoRTESum achieves a slight improvement of 0.51% in comparison to Sentence Lead [11] because the highlights were generated by the same mechanism of Sentence Lead, taking some first sentences and changing some keywords. We guess that the results will change when SoRTESum is evaluated on other datasets where the highlights are selected from the original document instead of abstract generation.

SoRTESum one wing obtains acceptable (outperforms Random and LexRank) showing the efficiency of our features. *Interwing-sent2vec* yields comparable results of ROUGE-1 on both sides indicating vector representation can be used for summarization in an appropriate manner. In *Dualwing-sent2vec*, however, the performance is decreased because many negative values appearing in vec-

tors leading the accumulative mechanism in Eq. (3) and (4) is inefficient. This suggests deeper investigations of calculation should be considered.

3.5 The Role of the Features

We further examined the role of the features by removing each feature and keeping the remaining ones using 1-fold validation. The results are shown in Table 4; italic is the statistical features.

Table 4: Top five effective features; * is Inclusion-exclusion coefficient

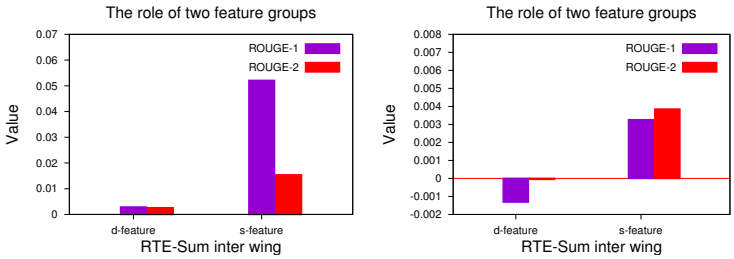
Feature	Document		Feature	Tweet	
	ROUGE-1	ROUGE-2		ROUGE-1	ROUGE-2
Overlap-coefficent	0.23 x 10^{-2}	0.25 x 10^{-2}	Euclidean	0.4 x 10^{-3}	0.3 x 10^{-3}
Dice coefficient	0.21 x 10^{-2}	0.14 x 10^{-2}	Dice coefficient	0.38 x 10^{-4}	0.203x10^{-4}
*In-ex coeffi**	0.7 x 10^{-3}	0.7 x 10^{-3}	*In-ex coeffi**	0.33x10^{-4}	0.203x10^{-4}
Jaccard	0.4 x 10^{-3}	0.5 x 10^{-3}	Jaccard	0.33x10^{-4}	0.203x$10^{-}4$
Matching coefficient	0.1 x 10^{-3}	0.3 x 10^{-3}	Manhattan	0.1x10^{-5}	0.4x10^{-5}

Both distance and statistical features affect the summarization. In a document, statistical features (in italic) play an important role. This shows that important sentences include important common words/phrases. On both sides, Dice coefficient, Inclusion-exclusion coefficient, and Jaccard have positive impact of the summarization. On the tweet side, however, distance features are more important than the remaining ones (only Inclusion-exclusion coefficient appearing).

The impact of *d-feature* and *s-feature* is illustrated in Figure 2. Statistical features have a positive impact on both sides in generating summaries (big value of 0.05 and round 0.004) whereas d-feature has negative influence in tweet summarization in Figure 2b. This concludes that *s-feature* plays an important role in document summarization. Although each distance feature in Table 4 has positive impact, combining them may lead to feature conflict. Note that negative values are very small (5.2x10^{-5}).

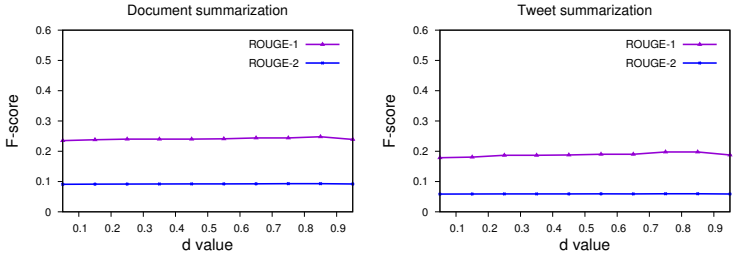
3.6 Tuning Damping Threshold

The impact of the damping factor in Eq. (3) and (4) was considered by adjusting $\delta = [0.05..0.95]$, changing value = 0.1. Results from Figure 3 show that when δ increases, auxiliary information benefits the performance of the model until some turning points. The performance is generally improved when δ closes to 0.85. After that, the performance slightly drops because with $\delta > 85$, the model is nearly same with *RTE one wing*. We therefore empirically selected $\delta = 0.85$. Note that the change is not much different among the tuning points because the RTE score is computed by averaging RTE features, hence the role of δ may be saturated.



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

Fig. 2: The impact of feature groups in our models



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

Fig. 3: Parameter adjustment of δ of RTE-Sum dual wing

3.7 Error Analysis

In Table 5 (the web interface can be seen at SoRTESum system[14]), both models yield the same results in document summarization, in which *S1*, *S2*, and *S3* are important sentences. Clearly, the content of these sentences completely relate to the highlights, which mention about the death of Tamerlan Tsarnaev at the Boston bombing event or attending information in his college. In contrast, *S4* mentioning his father information has light relevance.

In tweet summarization, the two methods generate the same three tweets and the remaining one is different. The summarization contains one the same tweet (*T1* in *SoRTESum inter wing* and *T3* in *SoRTESum dual wing*); the other ones are different making the difference of summarization performance between the two models. They are quite relevant to this event, but do not directly mention the death of Tamerlan Tsarnaev e.g., *T2*. This leads to lower performance for both models.

Finally, although social information can help to improve summary performance, other irrelevant data can badly affect the generation. This is because a score of an instance was calculated by an accumulative mechanism; therefore, common information (sentences or tweets) can achieve high score. For example, some tweets mention about the forgiveness of Tamerlan Tsarnaev's uncle e.g.,

[14] http://150.65.242.101:9293

Table 5: Summary example of our methods; bold style is important instances; [+] shows a strongly relevance and [-] is a light relevance.

Highlights	
+ HL1: Police identified Tamerlan Tsarnaev, 26, as the dead Boston bombing suspect	
+ HL2: Tamerlan studied engineering at Bunker Hill Community College in Boston	
+ HL3: 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	
RTE-Sum inter wing	**RTE-Sum dual wing**
+T1: Before his death Tamerlan Tsarnaev called an uncle and asked for his forgiveness. Said he 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**

T2. This, obviously, does not directly show the information of Tamerlan Tsarnaev's death, but this tweet received a lot of attention from readers when reading this event. More importantly, all sentences and tweets in Table 5 contain keywords that relate to Tamerlan Tsarnaev's event. This illustrates the efficiency of our method and suggests that the performance of the models can be improved based on informative phrases as *Generation* hypothesis in Section 1.

4 Conclusion

This paper presented *SoRTESum*, a summary framework using social information for summarization. Our framework utilizes a ranking approach to select important sentences and representative tweets in a novel similarity calculation. This paper also makes the contribution of formulating a sentence-tweet pair by RTE and proposes rich features to calculate the RTE score. Experimental results show that our approach achieves improvements of 0.51% to 8.8% over the unsupervised baselines and comparable results in comparison to supervised methods of ROUGE-1 in document summarization. In ROUGE-2, our models outperform all methods in tweet summarization.

For future direction, other important features e.g., NER or tree edit distance of the RTE task should be considered and integrated into the model. Another point is that our model should be compared to other supervised learning methods e.g., SVMs, CRF, or the model in [18,17]. An interesting point is that since deep learning has achieved promising results in NLP, we would like to adapt this technique to improve summary quality. Finally, abstract summarization should be considered in order to model the semantics in summarization.

Acknowledgment

We would like to thank to 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 to anonymous reviewers for their detailed comments for improving our paper. This work was partly supported by JSPS KAKENHI Grant number 3050941.

References

1. Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. Recognizing textual entailment: Rational, evaluation and approaches - erratum. *Natural Language Engineering 16(1): 105-105*, 2010.
2. Gunes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research, 22:457479*, 2004.
3. Wei Gao, Peng Li, and Kareem Darwish. Joint topic modeling for event summarization across news and social media streams. In *CIKM:1173-1182*, 2012.
4. Meishan Hu, Aixin Sun, and Ee-Peng Lim. Comments-oriented blog summarization by sentence extraction. In *CIKM: 901-904*, 2007.
5. Meishan Hu, Aixin Sun, and Ee-Peng Lim. Comments-oriented document summarization: Understanding document with readers' feedback. In *SIGIR: 291-298*, 2008.
6. Po Hu, Cheng Sun, Longfei Wu, Dong-Hong Ji, and Chong Teng. Social summarization via automatically discovered social context. In *IJCNLP: 483-490*, 2011.
7. Lifu Huang, Hongjie Li, and Lian'en Huang. Comments-oriented document summarization based on multi-aspect co-feedback ranking. In *WAIM: 363-374*, 2013.
8. Chin-Yew Lin and Eduard H. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *HLT-NAACL: 71-78*, 2003.
9. Yue Lu, ChengXiang Zhai, and Neel Sundaresan. Rated aspect summarization of short comments. In *WWW: 131-140*, 2009.
10. Hans P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research Development, 2(2): 159165*, 1958.
11. Ani Nenkova. Automatic text summarization of newswire: lessons learned from the document understanding conference. In *AAAI: 1436-1441*, 2005.
12. Minh-Tien Nguyen, Quang-Thuy Ha, Thi-Dung Nguyen, Tri-Thanh Nguyen, and Le-Minh Nguyen. Recognizing textual entailment in vietnamese text: An experimental study. In *KSE. DOI 10.1109/KSE.2015.23*, 2015.
13. Minh-Tien Nguyen, Asanobu Kitamoto, and Tri-Thanh Nguyen. Tsum4act: A framework for retrieving and summarizing actionable tweets during a disaster for reaction. In *PAKDD (2): 64-75*, 2015.
14. Martin F. Porter. Snowball: A language for stemming algorithms. 2011.
15. Xiaojun Wan and Jianwu Yang. Multi-document summarization using cluster-based link analysis. In *SIGIR: 299-306*, 2008.
16. Zhongyu Wei and Wei Gao. Utilizing microblogs for automatic news highlights extraction. In *COLING: 872-883*, 2014.
17. Zhongyu Wei and Wei Gao. Gibberish, assistant, or master?: Using tweets linking to news for extractive single-document summarization. In *SIGIR: 1003-1006*, 2015.
18. Zi Yang, Keke Cai, Jie Tang, Li Zhang, Zhong Su, and Juanzi Li. Social context summarization. In *SIGIR: 255-264*, 2011.