Using Facebook Public Posts to Enhance Trending News Summarization

Chen Li[1], **Zhongyu Wei**[3], **Yang Liu**[2], **Yang Jin**[2], **Fei Huang**[4]
[1] Microsoft, Bellevue, WA, USA
[2] Computer Science Department, The University of Texas at Dallas
[3] School of Data Science, Fudan University, Shanghai, P.R.China
[4] Facebook, 770 Broadway, New York, NY, USA
chei@microsoft.com {yangl,yangjin@hlt.utdallas.edu}
zywei@fudan.edu.cn feihuang@fb.com

Abstract

Summaries for trending news topics are often created from one or more news articles. In this paper we explore using relevant Facebook public posts in addition to the news articles to improve summarization of trending news. We propose different approaches incorporating information from public posts, including using frequency information from the posts to re-estimate bigram weights in the ILP-based summarization model and to re-weight a dependency tree edge's importance for sentence compression, directly selecting sentences from posts as the final summary, and finally a strategy to combine the summarization results generated from news articles and posts. Our experiments show that relevant Facebook public posts provide useful information and can be effectively leveraged to improve summarization results.

1 Introduction

Social networking services such as Facebook often provide lots of trending news topics and related news, comments, or posts for users everyday. In order to make information easier to digest, Facebook manually generates a short paragraph for each trending topic. Following that short paragraph, there are many public posts related to that news (relevant news or comments on the news). They may be published by news press like CNN, ESPN, and NBC etc., or by personal Facebook users. Usually, inside the post from a press, a URL of a related news article from that press is included as well. The human generated short paragraph is based on one of these news articles. An example of such a trending news summary and corresponding posts is shown in Fig1. Due to the space limit, we only show one following post which is on the right side in the picture.

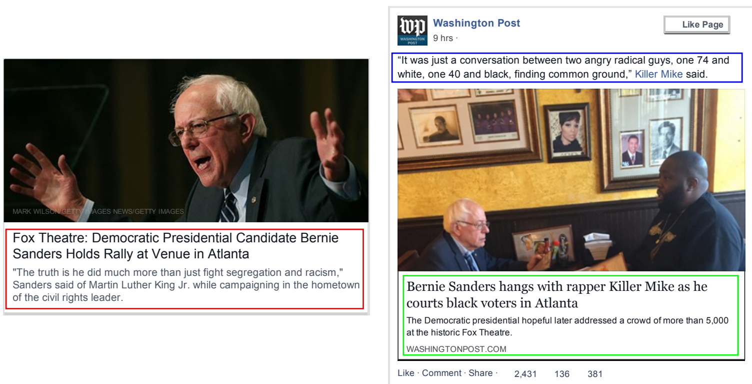

Figure 1: Facebook Trending News Topic Example. The short manual summary is marked in red rectangle. The blue rectangle shows a post from a user. In the green rectangle, it is a link of a related news. Some personal accounts may only publish posts without including related news.

Obviously, it is very time consuming to manually produce high quality summaries for many trending topics. Therefore, performing automatic summarization using the related news article is a straightforward method to alleviate the manual work. Since when a trending news is detected, we have access to not only the related news articles, but also the related public posts, our task in this paper is exploring how to use relevant public posts to improve summarization of trending news topics. This is also motivated by the following observations of the data. First, the posts under a trending news topic are closely related to and very indicative for that news. Second, by analyzing the post data,we find generally the sentences in posts are shorter with less irrelevant information than those from the news, thus they may be more suitable to be used as summary sentences. Third, the sentences from the posts whose accounts are maintained by news agencies are well written, so they can be directly used as the units of extractive summarization.

Our contributions in this paper are as follows: (1) We propose an integer linear programming (ILP) based trending news summarization approach on Facebook trending news service. It involves generating extractive and abstractive summaries. (2) We explore various ways of using post information to boost summarization performance. There are 3 general strategies: one is to leverage the lexical frequency information in the post to help estimate a word's importance in the news article and thus choose better summary sentences; another one is to extract sentences from the posts to form the summary; and the last one is to combine summarization results generated from the news articles and the posts. (3) To evaluate our method, we collect 190 trending news topics from Facebook. Each one has a news article, a human generated summary and hundreds to thousands of related public posts. To our knowledge, this is the first Facebook based social media summarization data set.

2 Related Work

Our work is closely related to the following aspects: ILP based summarization method, dependency tree based sentence compression by considering extra information, and mining social media for document summarization.

Recently optimization methods have been widely used in extractive summarization. McDonald (2007) first introduced sentence level ILP for summarization. Later Gillick et al. (2009) revised it to concept-based ILP, which is similar to the Budgeted Maximal Coverage problem in (Khuller et al., 1999). Then the optimization method is widely used in summarization task (Lin and Bilmes, 2010; Davis et al., 2012; Li et al., 2015b; Li et al., 2015a). In such ILP-based methods, how to determine the concepts and measure their weights are the two key factors impacting the system performance. Woodsend and Lapata (2012) utilized ILP to jointly optimize different aspects including content selection, surface realization, and rewrite rules in summarization. Galanis et al. (2012) uses ILP to jointly maximize the importance of the sentences and their diversity in the summary. In this work, we leverage the unsupervised ILP framework from Gillick et al. (2009) as our summarization system and incorporate post information to help boost summarization performance.

Sentence compression techniques are widely used in summarization in order to generate abstractive summaries. Previous research has shown the effectiveness of sentence compression for automatic document summarization (Knight and Marcu, 2000; Zajic et al., 2007; Chali and Hasan, 2012; Wang et al., 2013). The compressed summaries can be generated through a pipeline approach that combines a generic sentence compression model with a summary sentence pre-selection or post-selection step. In addition, joint summarization and sentence compression method attracts lots of attention these years. (Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011; Li et al., 2014) are typical work in this area. Their focus is to leverage the ILP technique to jointly select and compress sentences for multi-document summarization. In our work, we consider posts as summary related information and then use them for joint sentence compression and summarization.

Although there is little work about generating summaries by considering extra information on Facebook data, there is some similar work done on Twitter or other resources. Unsupervised method was tried on summarization task by (Wong et al., 2008). (Phelan et al., 2011) used tweets to recommend news articles based on user preferences. (Gao et al., 2012) produced cross-media news summaries by capturing the complementary information from both sides. Kothari et al. (2013) and Štajner et al. (2013) investigated detecting news comments from Twitter for extending news information provided. Wei and Gao (2014) derived external features based on a collection of relevant tweets to assist the ranking of the original sentences for highlight generation. In addition to tweets, Svore et al. (2007) leveraged Wikipedia and query log of search engines to help document summarization. Tsukamoto et al. (2015) proposed a method for efficiently collecting posts that are only implicitly related to an announcement post, taking into account retweets on Twitter in particular. Our work involves the two aspects when using post information: one is that we utilize post information to help choose sentences from new articles and compress them to form a summary, and the other is that we directly use sentences from the posts as the summary.

3 Corpus Construction

Since our work is based on Facebook trending news service and there is no public corpus from it, we manually collected them from Facebook from Oct 20, 2015 to Nov 10, 2015. During that time, we collected the top 10 trending news topics (each topic includes a human generated summary, a related news article and all the following posts). These topics may come from politics, science and sport categories. Since the trending news is dynamically updated (sometimes fast sometimes slow), we only collected once everyday to avoid repetitions. In order to better evaluate the impact of the relevant posts, we ignore the trending news whose post number is less than 50. In total, we collected 190 trending news topics and they only contains public posts[1].

The statistics of this corpus are given in Table 1. As shown in the table, the average number of relevant posts for a trending topic is about 217. The standard deviation of the number of posts associated with a topic is 188. The high variance is because some of the topics are much more popular than others. We also find that the average sentence length from posts (12 tokens) is about half of that from news (21 tokens), but the number of posts is much larger than that from news and we expect they can provide a very useful and indicative information to guide the summary extraction. The average summary length for each topic is 43 words. This means that a summary can only contain average two sentences or sometime just one long sentence from the news. But usually such one or two sentences can not represent all the information in the summary, therefore we may need to compress the long sentences or extract shorter sentences from the posts that contain similar information as the long sentences in the news.

	All	Politic	Science	Sports
# of Trending Topic	190	49	71	70
	News			
# of sent/news	22.83±13.27	27.71±15.91	19.94±12.13	22.35±11.23
# of token/sent	20.76±11.00	20.69±11.35	20.96±10.59	20.65±11.05
	Posts			
# of post/topic	216.45±187.56	299.04±262.56	236.58±166.81	138.23±87.77
# of sent/topic	454.28±468.54	459.01±280.48	725.08±753.52	259.93±171.71
# of token/sent	12.85±11.14	11.66±10.61	14.45±11.89	11.87±10.16
	Summary			
# of token/topic	43.34±4.76	43.68±4.37	42.22±4.39	44.22±5.14
# of token/sent	21.67±8.98	21.84±8.47	21.11±8.96	22.11±9.3

Table 1: Overview statistics on the corpus (mean and standard deviation)

4 Extractive Summarization Methods and Results

4.1 ILP Based Document Summarization

The core idea of using ILP for summarization is to select the summary sentences by maximizing the sum of the weights of the language concepts that appear in the summary. Gillick et al. (2009) showed that using bigrams as concepts gave consistently better performance than unigrams or trigrams for a variety of ROUGE measures. The association between the language concepts and sentences serves as the constraints. This ILP method is formally represented as below:

[1]The data is available at http://www.hlt.utdallas.edu/∼chenli/summarization

$$
\begin{aligned}
max \quad & \textstyle\sum_i w_i b_i && (1)\\
s.t. \quad & s_j Occ_{ij} \leq b_i && (2)\\
& \textstyle\sum_j s_j Occ_{ij} \geq b_i && (3)\\
& \textstyle\sum_j l_j s_j \leq L && (4)\\
& b_i \in \{0,1\}\ \forall i, \ \ s_j \in \{0,1\}\ \forall j && (5)
\end{aligned}
$$

b_i and s_j are binary variables that indicate the presence of a bigram and a sentence respectively. l_j is the sentence length and L is maximum length of the generated summary. w_i is a bigram's weight and Occ_{ij} means the occurrence of concept i in sentence j. Inequalities (2) and (3) associate the sentences and concepts. They ensure that selecting a sentence leads to the selection of all the concepts it contains, and selecting a concept only happens when it is present in at least one of the selected sentences.

4.2 Generating summaries from news article

In this setup we extract sentences from news articles using the ILP based summarization framework. Our main goal is to investigate if we can use the relevant posts to better determine the bigrams and their weights in the ILP model described above. We compare the following three methods to choose and weight bigram:

- **Bigram and Weight from News Article**: we use the bigram in the news article and its augmented term frequency as its weight: $w_i = 0.5 + \frac{f_{i,d}}{max\{f_{i,d}:i \in d\}}$ ($f_{i,d}$ is the raw frequency of bigram i in document d).

- **Bigram from News and Posts**: among the bigram candidates extracted from the news article, we use the subset that also appear in the posts, and the same weight as above (that is, the weight information is just based on the news article).

- **Bigram and Weight both from News and Post**: using the common bigrams from both the news article and posts (same as the previous setup), we further update the bigram weight by adding a bigram's post frequency in the relevant posts. In the following equation , pf_i is the number of posts that contain bigram i: $w_i^{'} = 0.5 + \frac{f_{i,d}}{max\{f_{i,d}:i \in d\}} + pf_i$.

4.3 Generating summary from posts only

In this setup, we evaluate whether sentences from posts are good candidates for a summary. Here each post can be seen as an individual document and we can treat this as a ‘multi-document’ summarization task and easily apply the ILP module on all the posts to choose a set of sentences as the final summary. In this process, the input sentences and bigrams are only from the posts, and the bigram weight is post frequency: $w_i^{''} = pf_i$ (Number of posts the bigram has appeared.).

4.4 Generating summary from news and posts

Here we use all the sentences from the news and posts as the input for summarization. This is again a multi-document summarization task, where we consider each post and the new article as a document. The bigram weight is document frequency. This method combines the news article and posts together to form a document collection for summarization. In the following we call it document level combination.

4.5 Combination of summarization results from news article and posts

In contrast to the above combination method, we can also build summarization systems using the news article and the relevant posts separately, and then combine the generated summaries. This kind of summary result level combination allows us to develop individual models tailored for different input sources, and may produce better combined final results. In this combination method, we have two summarization results, generated from the sentences in the news article and the posts respectively, our aim is to decide which of the two summaries is better and use that as the final result. Since we do not have enough data to train supervised models, we propose to use heuristic rules to select which summary to use. The combination rules are based on the following parameters.

- **Sentence number**: $n_{sentNum}$. This represents how many sentences a summary result consists of. We observe that often when a summary contains just one sentence, that sentence is the news highlight and contains the most important information.
- **Bigram weight**: w_i in Section 4.1. Sentences containing bigrams with high weights are often good summary sentence candidates. We further define $w_{maxInTopic}$ as the maximum weight of the bigram in a topic, and $w_{maxInRes}$ as the maximum weight of the bigram in the summary result.
- **Bigram exist ratio**: R_{Bigram}, which represents the percentage of bigrams in a sentence that are used as variables in the ILP formula.We define this ratio since we prefer sentences that contain more bigrams that are used in the ILP model.

Then our rule-based classifier works by going through the following rules one by one. If a decision can be made at any point, the procedure will stop.

- **Rule 1**: If $n_{sentNum}$ from a summary result equals to one and the length of that sentence is longer than 40 words, choose that result. If both or neither equals to one, go to Rule 2.
- **Rule 2**: If $w_{maxInRes}$ from the post summary equals to $w_{maxInTopic}$, but if it is not true for the summary from news, choose the result from posts as the final summary. Otherwise, go to Rule3.
- **Rule 3**: If the maximum R_{Bigram} from a sentence in post result is larger than a threshold value[2], use the post result as the final summary; otherwise use the news result as the final summary. If the maximum R_{Bigram} from post and news results are the same, go to Rule 4.
- **Rule 4**: Choose the result with higher average R_{Bigram}. If average R_{Bigram} are the same, go to Rule 5.
- **Rule 5**: Choose news result as the final result.

4.6 Results

The summary length is set as 45 words maximum (because the average length of human summary is 43 words in each topic). The compared methods include:

(a) Summary sentences from news article I: bigrams are from news, and weight is their augmented term frequency from news.

(b) Summary sentences from news article II: bigrams are from both news and posts, and weight is their augmented term frequency from news.

(c) Summary sentences from news article III: bigrams are from news, and weight is the combination of their augmented term frequency from news and their raw post frequency.

(d) Summary sentences from news article IV: bigrams are from news and posts, and weight is the combination of their augmented term frequency from news and their raw post frequency.

(e) Summary sentences from posts: bigrams are from posts, and weight is their post frequency.

(f) Document level combination: sentences are from news or posts, and bigram weight is ‘document’ frequency.

[2]This value is empirically set as 0.85 in our experiments.

(g) Summary result level combination: given two summaries with sentences extracted from either news or posts, decide which one to use as the final result.

A sentence in the post may be exact same with a sentence in human summary. One possible reason is that the users may first read the summary and then publish the post by copying that summary. In order to minimize this effect, we only consider the posts whose cosine similarity with corresponding human summary is lower than 65%. For measurement, we use the ROUGE evaluation metrics (Lin, 2004) with R-1 and R-2 measuring the unigram and bigram overlap between the system and reference summaries, and R-SU4 measuring the skip-bigram with the maximum gap length of 4. Table 2 presents the recall performance of these systems in ROUGE-1, ROUGE-2 and ROUGE-SU4 along with corresponding 95% confidence intervals. We determine which differences in scores are significant by comparing the 95% confidence intervals, significant differences are those where the confidence intervals for the estimates of the means for the two systems either do not overlap at all, or where the two intervals overlap but neither contains the best estimate for the mean of the other.

From the results we find that systems using only information from the news (e.g., 'a') performs worst. This also shows that this kind of single document summarization is not a trivial task. After adding information from posts, such as requiring the bigrams to also appear in posts (system 'b') or computing bigram weights using post related frequency (system 'd'), the results (system 'd' compared with 'a' and 'b') improved significantly. It is consistent with our expectation that post information can help enhance summarization of trending news topics.

System	ROUGE-1	ROUGE-2	ROUGE-SU4
a	0.30650 (0.29449 - 0.31896)	0.08621 (0.07620 - 0.09627)	0.10776 (0.09996 - 0.11737)
b	0.35453 (0.34173 - 0.36710)	0.12304 (0.11172 - 0.13474)	0.13940 (0.12948 - 0.14956)
c	0.37459 (0.36327 - 0.38507)	0.13655 (0.12698 - 0.14593)	0.14746 (0.13935 - 0.15554)
d	0.37943 (0.36838 - 0.39157)	0.14359 (0.13328 - 0.15548)	0.15391 (0.14503 - 0.16425)
d oracle	0.42377 (0.41130 - 0.43573)	0.21249 (0.20051 - 0.22445)	0.19915 (0.18825 - 0.21047)
e	0.39787 (0.38695 - 0.40930)	0.16292 (0.15314 - 0.17323)	0.16596 (0.15778 - 0.17464)
e oracle	0.54269 (0.53003 - 0.55503)	0.34810 (0.33195 - 0.36409)	0.31372 (0.29901 - 0.32948)
f	0.39182 (0.38048 - 0.40369)	0.15504 (0.14436 - 0.16643)	0.16359 (0.15489 - 0.17349)
g	0.40651 (0.39526 - 0.41793)	0.17254 (0.16178 - 0.18408)	0.17499 (0.16566 - 0.18532)

Table 2: Recall of ROUGE-N results on different extractive summarization systems.

One important finding from Table 2 is the results from post sentences (system 'e') are even better than that from news article sentences. To better understand this, we conducted an oracle experiment when extracting sentences from the news article and posts respectively: we use the bigrams from the reference summary as the bigram concepts in the ILP method, and the weight is the bigram's term frequency in the reference summary. This oracle experiment can reflect the possible best result of the extractive summarization system when extracting sentences from news or posts. The results are also included in Table 2. We can see that the possible best summaries from posts are also significantly better than that from news. By analyzing the results of this oracle experiment, we find that the average length of the generated summary from news is 38.15 tokens and the length of summary from posts is 41.25. It meas the summary generated from posts contains more information. In another aspect, the summary from news only contains 2.1 sentences in average, but this number in the summary from posts is 2.5. As mentioned earlier, the sentences from posts are typically shorter than those from news. Therefore when the target summary has short length limit (for example 45 tokens, usually less than 3 sentence), one informative long sentence could use up all the length budget while shorter sentences allow various information to be incorporated. Similar pattern is also found in the results of system d and system e. The average length of summary from system d is 41.8 and there are 2.3 sentences in average while the corresponding number from system e is 43.4 and 2.7. After taking a look at the summary from system e, we also find the selected short sentences indeed include more than one aspects of the topic, whereas the long sentences which usually only contains one aspect of the topic.

Even though overall system 'e' has the best performance, after analyzing the results, we find only for 110 topics, the summary results from the posts are better than that from the news article, and for the remaining 80 topics, the results based on the news sentences are better. This also justifies why we expect combining results from the news and the posts based summaries may improve system performance. From the results in Table 2, we find that document level combination (system 'f') is not very effective. It is similar to the results using just the posts. A better bigram selection and weighting strategy may be needed when combining the posts and news at the input level. However, summary result level combination (system 'g') significantly outperforms each individual system, suggesting we can build each individual system, and then effectively choose one as the final output. The oracle result combination (i.e., comparing to the reference summary and picking the one with better scores as the system prediction) has a ROUGE-2 Recall score of 0.1922 (0.18085 - 0.20394). Our rule based combination method is quite close to the oracle combination result, indicating our rules can measure the goodness of a system generated summary.

5 Abstractive Summarization Method and Results

5.1 Dependency Tree Based Compression

We have mentioned that sentences from the news are generally long. Intuitively compressing the sentences in the news will give us room to incorporate more information. In fact, we noticed that the summaries generated from the news sentences are on average shorter than that from the posts. This is due to the use of long sentences and meeting the summary length constraint. Therefore next we investigate abstractive summarization by applying sentence compression when extracting sentences from news to improve summarization performance. Here the core idea of our proposed compression method is using the information from revelent posts to guide compression. Our compression framework is inspired by the work in (Filippova and Strube, 2008), where they use extra resources to guide the unsupervised dependency tree based sentence compression module.

The sentence compression task can be defined as follows: given a sentence s, consisting of words $w_1, w_2, ..., w_m$, identify a subset of the words of s, such that it is grammatical and preserves essential information of s. In the framework of (Filippova and Strube, 2008), a dependency graph for the original sentence is first generated and then compression is done by deleting edges of the dependency graph. The goal is to find a subtree with the highest score:

$$max \sum_{e_i \in E} a_{e_i} * w_{info}(e_i) * w_{syn}(e_i) \tag{6}$$

where a_{e_i} is a binary variable, indicating whether a directed dependency edge e_i is kept (a_{e_i} is 1) or removed (a_{e_i} is 0), and E is the set of edges in the dependency graph. The weighting of edge e considers both its syntactic importance ($w_{syn}(e_i)$) and the informativeness ($w_{info}(e_i)$). Suppose edge e_i is pointed from head h to node n with dependency label l, we use two methods to calculate the two weights in Formula 6.

The first one uses a bigram news corpus with the corresponding summaries. In formula $w_{info}(e_i) = \frac{P_{summary}(n)}{P_{news}(n)}$ and $w_{syn}(e_i) = P(l|h)$, $P_{summary}(n)$ and $P_{news}(n)$ are the unigram probabilities of word n in the language models trained on human generated summaries and the original news articles respectively. $P(l|h)$ is the conditional probability of label l given head h. We used the New York Times Annotated Corpus (LDC Catalog No: LDC2008T19) as the extra background corpus. It has both the original news articles and human generated summaries.

In the second method, we explore using relevant posts as background information for compression. In $w_{info}(e_i) = \frac{pf(n)}{\#Post}$ and $w_{syn}(e_i) = \frac{pf(h,n)}{\#Post}$, $pf(n)$ is the number of posts including word n and $pf(h, n)$ is the number of posts where n and head h appear together. If h and n appear together in two sentences in one post, it is counted as one. #Post represents the total number of posts in a topic.

5.2 Joint model for summarization and sentence compression

We propose a joint model for sentence selection and compression at the same time under the ILP framework, in order to avoid the problem with pre-compression (error propagation due to imperfect compression, important information may be missing) or post-compression (after compression it is hard to add new sentences to use the new available space). In the joint model, we combine the objectives in Section 4.1 and Formula 6, and thus the goal is to find a set of sentences with the highest score:

$$max \sum_{e_{jk} \in E} \lambda * a_{e_{jk}} * w_{info}(e_{jk}) * w_{syn}(e_{jk}) \ + \sum_{i} w_i b_i, \quad \forall i, j, k \tag{7}$$

e_{jk} means the k^{th} edge in j^{th} sentences in this news article. λ is used to balance the contribution from the edge importance and bigram weights. After we add edges into our ILP-based summarization model, we need to adjust the previous constraints and also design more constraints to represent relationships between sentences and edges, bigrams and edges in order to produce valid results.

First, the length constraint in Section 4.1 should be expressed in the form of edges rather than sentences.

$$\sum_{j,k} a_{e_{jk}} \leq L - 1, \ \forall j, k \tag{8}$$

Second, we want to avoid picking just a few words from many sentences as the summary, which typically leads to ungrammatical summaries. Hence it is more desirable to obtain a solution with only a few sentences extracted and compressed. To do so, we create the relationship between edges and sentences like following: if sentence j is selected, there are at least $\rho * L_j$ words extracted. L_j is the length of sentence j. This constraint is shown in the first inequality in Formula 9. Combined with the second inequality in, they together make sure that if sentence j is selected, at least $\rho * L_j$ words will be chosen; if sentence j is not selected, none of edges from this sentence will be selected.

$$\sum_{j,k} a_{e_{jk}} \geq \rho * L_j * s_j, \quad a_{e_{jk}} \leq s_j, \ \forall j, k \tag{9}$$

Third, one bigram has two tokens, meaning it involves at least one edge and at most two edges. Therefore we build the relationship between bigrams and edges as follows:

$$b_i \geq a_{e_{jk}}, \qquad b_i \leq \sum a_{e_{jk}} \tag{10}$$

where e_{jk} represents all the edges whose head h or node n is one element of bigram i.

Forth, in the dependency tree, if an edge $e_{j,k}$ is removed, all the edges whose head node is $e_{j,k}$'s node n need to be removed as well.

$$a_{e^l_{jk}} \geq a_{e^{l+1}_{jk'}} \tag{11}$$

in which edge $e^{l+1}_{jk'}$ is at level $l+1$ and its head node is the node n of e^l_{jk} at level l. Please note we do not include the vice verse constraints which means even if all the edge $e^{l+1}_{jk'}$ are removed, we can still keep edge e^l_{jk}.

In addition to all the constraints from Formula 7 to 11, we require that b_i, s_j and $a_{e_{jk}}$ are all binary variables. This gives the ILP setup for the joint summarization and sentence compression model.

5.3 Results

The abstractive summarization experiments are based on the setup of System 'd', that is, we extract sentences from the news articles, but the bigrams and their weight information come from both the news and the posts. We use the joint summarization and compression method described above, with extra background information to help guide compression. λ in Formula 7 and ρ in Formula 9 are empirically set as 20 and 0.85 respectively in our experiment.

Results are shown in Table 3. For System 'd', we present results using two different resources for compression: the generic NY Times Corpus and the relevant posts for each topic. We find adding compression improves summarization performance over the extractive summarization baseline. Using posts as extra information outperforms that using the general news. This improvement is also statistically significant. In the table we also include the result using the System 'g' configuration. For this method, once the combination rules determine to use the extractive summary from the news as the final system output, we apply abstractive summarization (i.e., joint compression and summarization) to this topic to regenerate the summary. We can see applying compression on these topics gave another improvement over the original combination result.

Compression System		ROUGE-1	ROUGE-2	ROUGE-SU4
Based on	Extra Resource			
Sys d	NYT corpus	0.40437 (0.39326 - 0.41586)	0.15059 (0.14095 - 0.15985)	0.16311 (0.15484 - 0.17167)
	Post	0.41111 (0.40025 - 0.42282)	0.15567 (0.14561 - 0.16637)	0.17100 (0.16231 - 0.18051)
Sys g	Post	0.41232 (0.40133 - 0.42329)	0.17495 (0.16421 - 0.18653)	0.17871 (0.16983 - 0.18879)
Extractive System (d)		0.37943 (0.36838 - 0.39157)	0.14359 (0.13328 - 0.15548)	0.15391 (0.14503 - 0.16425)

Table 3: Recall of ROUGE-N results on abstractive summary.

6 Conclusion and Future Work

In this paper we explore utilizing relevant Facebook public posts in addition to the news articles for summarization of trending news. We adopt the ILP based summarization method and propose different ways using information from posts, including weighting the bigrams using frequency information from the posts, compressing news sentences by estimating importance of dependence tree edges based on occurrence information in the posts, selecting sentences from posts as final summary, and finally combining the results generated from news articles and posts. Our experiments show that post information is useful for improving the performance.

We plan to pursue a number of directions in our future work. First, we plan to use a statistical classifier to choose a better summary for system combination. Second, we will perform more fine grained combination by choosing individual sentences from different results. Third, we will investigate multi-document summarization once we can collect multiple news articles for a trending topic.

Acknowledgements

References

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of ACL.*

Yllias Chali and Sadid A. Hasan. 2012. On the effectiveness of using sentence compression models for query-focused multi-document summarization. In *Proceedings of COLING*.

Sashka T. Davis, John M. Conroy, and Judith D. Schlesinger. 2012. Occams - an optimal combinatorial covering algorithm for multi-document summarization. In *Proceedings of ICDM*.

Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*.

Dimitrios Galanis, Gerasimos Lampouras, and Ion Androutsopoulos. 2012. Extractive multi-document summarization with integer linear programming and support vector regression. In *Proceedings of the COLING*.

Wei Gao, Peng Li, and Kareem Darwish. 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*. ACM.

Dan Gillick, Benoit Favre, Dilek Hakkani-Tur, Berndt Bohnet, Yang Liu, and Shasha Xie. 2009. The ICSI/UTD summarization system at tac 2009. In *Proceedings of TAC*.

Samir Khuller, Anna Moss, and Joseph Seffi Naor. 1999. The budgeted maximum coverage problem. *Information Processing Letters*.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *AAAI*.

Alok Kothari, Walid Magdy, Ahmed Mourad Kareem Darwish, and Ahmed Taei. 2013. Detecting comments on news articles in microblogs. In *Proceedings of ICWSM*.

Chen Li, Yang Liu, Fei Liu, Lin Zhao, and Fuliang Weng. 2014. Improving multi-documents summarization by sentence compression based on expanded constituent parse trees. In *Proceedings of EMNLP*.

Chen Li, Yang Liu, and Lin Zhao. 2015a. Improving update summarization via supervised ilp and sentence reranking. In *Proceedings of the NAACL*.

Chen Li, Yang Liu, and Lin Zhao. 2015b. Using external resources and joint learning for bigram weighting in ilp-based multi-document summarization. In *Proceedings of NAACL*.

Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Proceedings of NAACL*.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Proceedings of ACL*.

Andre F. T. Martins and Noah A. Smith. 2009. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the ACL Workshop on Integer Linear Programming for Natural Language Processing*.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of ECIR*.

Owen Phelan, Kevin McCarthy, Mike Bennett, and Barry Smyth. 2011. Terms of a feather: Content-based news recommendation and discovery using twitter. In *Advances in Information Retrieval*. Springer.

Tadej Štajner, Bart Thomee, Ana-Maria Popescu, Marco Pennacchiotti, and Alejandro Jaimes. 2013. Automatic selection of social media responses to news. In *Proceedings of the 19th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*. ACM.

Krysta Marie Svore, Lucy Vanderwende, and Christopher JC Burges. 2007. Enhancing single-document summarization by combining ranknet and third-party sources. In *Proceedings of EMNLP-CoNLL*.

Yuma Tsukamoto, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2015. Collecting microblog posts implicitly related to announcement post. In *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A sentence compression based framework to query-focused multi-document summarization. In *Proceedings of ACL*.

Zhongyu Wei and Wei Gao. 2014. Utilizing microblogs for automatic news highlights extraction. COLING.

Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of COLING*.

Kristian Woodsend and Mirella Lapata. 2012. Multiple aspect summarization using integer linear programming. In *Proceedings of EMNLP-CoNLL*.

David Zajic, Bonnie J. Dorr, Jimmy Lin, and Richard Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. In *Information Processing and Management*.