A Novel Method for Summarization and Evaluation of Messages from Twitter

Conference Paper · December 2017

DOI: 10.1109/ICAPR.2017.8592951

Surender Singh Samant
BITS Pilani, Hyderabad
7 PUBLICATIONS **55** CITATIONS

Aruna Malapati
BITS Pilani, Hyderabad
43 PUBLICATIONS **508** CITATIONS

A Novel Method for Summarization and Evaluation of Messages from Twitter

Surender Singh Samant, N. L. Bhanu Murthy, Aruna Malapati

Department of Computer Science and Information Systems
Birla Institute of Technology and Science, Pilani
Hyderabad Campus, Telangana, India
Email: {surender.samant, bhanu, arunam}@hyderabad.bits-pilani.ac.in

***Abstract*—Users of microblogs such as Twitter publish messages about many topics including real world events. Patterns can be identified and used to classify events from these messages. It is challenging to summarize the event related messages concisely to get important information about the event. In this paper, we propose a novel method of event summarization that uses the frequently occurring sets of words and features of events related messages in a cluster. For effective evaluation of system summaries, we created a set of extensive reference summaries from manual summaries. These reference summaries were used to evaluate various summarisers using standard ROUGE based metrics. We compared our summarizer with many baselines on a set of 50 manually labelled and extended event summaries. In our experiments, the proposed summarizer outperformed other summarizers on ROUGE-1, ROUGE-2, and ROUGE-L.**

***Index Terms*—event summarization, pattern recognition, twitter**

1. Introduction

Social media has become a dominant medium for information sharing in the last few years. Users have begun to rely on microblogging platforms to publish or receive news. Twitter is the most prominent microblogging site. It is a heavy volume platform where millions of messages are published every hour. Many of these messages contain information and discussions about real world events. Since Twitter provides an API to track messages by keywords and download them in real time, it has become an interesting source for getting data for research. Message streams of interest can be collected and processed to identify patterns related to a real world event. Various methods have been proposed to collect messages, create clusters, identify patterns and classify events. As a simple example, hashtags in Twitter can be used to track messages of interest. There are also complex machine learning approaches (e.g. [1], [2]) that track the stream of messages to identify patterns of events. These methods can be used to classify a cluster of similar messages as event or non-event cluster. The task of event summarization is the next important step that presents important information about the event to the user. The information normally consists of one or more of the entities involved, action, location of event, etc.

Summary of an event is a concise text that provides most of the information about the event to the user. An interesting application of event summarization is to provide news alerts to users automatically by identifying events from a stream of twitter messages (tweets). An informational and concise summary is needed for good end user experience, and the overall performance of the whole system is often reflected in the system generated summary.

A summarizer (the method or algorithm) is either extractive where phrases from the messages are used to create summary, or abstractive where new phrases may be constructed from the messages. Due to inherent problems of natural languages, extractive summarization is the more popular approach since it uses already existing phrases. Abstractive summarization is susceptible to generating incoherent summaries when done automatically. Since Twitter messages are limited in length and summarization involves a cluster of messages that discuss the same topic or event, extractive methods have been effectively used to create a summarizer. These algorithms attempt to extract the most representative message as the summary of the event. It is challenging to extract such a message as the message cannot be too small as it may miss information, and not too big as that may introduce noise words that are not informational.

Table 1 contrasts a set of effective summaries extracted from Twitter with another set of non-effective summaries that are often lacking information or are completely irrelevant. It is clear that a good summary is concise, but long enough to provide most of the important information about the event. Since these requirements are subjective, there are multiple humans each of whom create a summary (called model summary) for each event cluster that is used to compare and evaluate the system extracted summary (called system or peer summary). This leads to the problem statement for this research:

Problem Statement: Given a cluster of event related messages, extract the message that is most representative of the event. The representative message should be concise but should provide the most information about the event.

Our main contributions in this research are: We present a novel method of summarizing event-related cluster of tweets that uses frequent sets of words along with the presence of frequent capitalized and action words (section 3.2). We created a set of model summaries for 50 clusters of events to evaluate our summarizer against multiple baselines. Each model summary was extended to create multiple reference summaries using Wordnet [3] to deal with a limitation of ROUGE metrics (discussed in section 4.1) . In multiple experiments, our proposed summarizer performed better than all the other algorithms (section 4.2).

2. Related Work

Event identification from short messages and summarization of event related messages is an active area of research. An unsupervised abstractive method was introduced in [4] to generate concise phrases from opinions. They modelled summarization as an optimization problem. The method generated key phrases that were not necessarily part of the text to be summarized. They used a web-based corpus to give weight to relevant phrases to be generated. Our goal is to extract the most representative message from a cluster of tweets that is different from this work. A Hidden Markov Model based method to summarize recurring sporting moments was introduced in [5]. They used the model to identify a series of pre-defined key moments during the sporting event. Sporting event summarization was also the topic of [6] that used spikes in the number of messages during such events as temporal cues to identify moments to summarize the event. The research in [7] used temporal correlation to create topic models to extract summary. Different from these methods, we use just the text of the messages to extract summary. Summarization of multiple documents using a combined tf-idf and frequent itemset mining based approach was performed in [8] where the goal was to extract the most representative and mutually non-redundant sentences from multiple documents such that the overall recall was maximized. Our work extracts a single representative message from multiple messages by weight assignments without tf-idf. The research in [9] has similar goal to our work. They proposed two methods: the first one used a phrase reinforcement (PR) approach that uses the overlapping sequence of words surrounding a key phrase to assign weights to messages and extract the summary. The second method which performed better used a variation of tf-idf to assign weights to messages and select the highest weighted sentence. We used the second method as one of the strong baselines to compare with our summarizer.

3. Summarization Methods

Among the many summarization methods that exist to summarize short texts, we used a few naïve baselines and two stronger ones, a centroid based summarizer and a hybrid tf-idf from [9]. These methods are explained next, followed by our proposed summarizer (OurSum).

FIFA to expand World Cup to 48 teams in 2026.	All World Cup group games decided in penalties!? Surely that's banter? What a joke.	FIFA announced that football world cup finals will be played between 48 nations in 2026.
Hawaii judge extends halt on Trump travel ban.	Wall fail, Obama care fail, travel ban fail. Talk about lame duck	Hawaii judge extends national block of Trump travel ban.
India wins Kabaddi World Cup 2016.	Yessssssssssssss,...... We Win.. World Cup...	India defeats Iran to win kabaddi world cup 2016.
A)	**B)**	**C)**

TABLE 1. EACH ROW CONTAINS THE FOLLOWING TYPES OF SUMMARY: A) INFORMATIONAL, B) NON-INFORMATIONAL, C) MANUAL

3.1. Baseline methods

Shortest, Longest, and Random summarizers: These are naïve methods that extract the shortest, longest, and a random message respectively as system summary. These methods are important to compare a summarizer against these naïve approaches.

Centroid-based summarizer: The centroid is computed and the nearest message to the centroid is selected as summary. This is better than the naïve summarizers and extracts reasonably good summary.

Hybrid TFIDF (H-tfidf) summarizer: This summarizer uses a modified tf-idf based approach to assign weights to each of the messages, and extracts the message with the most weight as summary of the cluster of messages. We implement the method as described in [9] and used the best performing parameter (normalization factor) computed empirically for our dataset.

3.2. Proposed summarizer

The key observation used to extract a message as system summary is that the sets of frequently used common words in a cluster of messages are very informative and this informativeness is directly proportional to the size of the set. A message that contains a bigger frequent set of words gives more information about the event than a message that contains a smaller set. So, the messages containing these frequent set of words should have more probability to be selected as the system summary than other messages. It should be noted that the frequent set of words is more general than a subsequence since the subsequence follows an ordering of words unlike a set. For example, the following two messages contain a frequent set of words {paris, attack, terrorists} but in completely different order.

M1: paris under attack by terrorists.

M2: terrorists attack paris.

Hence, the set of frequent words can identify similarity between two messages better than a subsequence if the messages use same set of words but in different order.

We also used some characteristics of English text to augment the weight of informative messages further. Commonly used capitalized words in a cluster of messages are often named entities and commonly used verbs in messages often signify an action performed by or on a named entity [10].

Our summarizer assigns weight to each message to identify the top weighted messages in a cluster. We used the following approach to computed weights:

Let S be the set that consists of all the frequent set of words in a cluster and $S_{max}(M) \in S$ be the frequent set of maximum size contained in M, then intuitively the weight w_M contributed to the message by S should be directly proportional to the size of $S_{max}(M)$ ($|S_{\max}(M)|$). In other words, given two messages M and M' with $S_{max}(M)$ and $S_{max}(M')$ as their respective maximum size frequent sets, then:

$w_{\mathrm{M}} > w_{\mathrm{M}'}$ if $|S_{\max}(M)| > |S_{\max}(M')|$, and vice versa.

As an example, if S = {{india, kabaddi, world, cup}, {india, world, cup}, {world, cup}}, and we have the following two messages (lowercased):

M_1: india won the kabaddi world cup 2016.

M_2: india lifts world cup.

$|S_{max}(M_1)|$ is 4, whereas $|S_{max}(M_2)|$ is 3, so M_1 should get more weightage. We set $w_M = k * |S_{\max}(M)|$ where k is a constant.

The top T capitalized words and verbs in a cluster contribute weights w_{cap} and w_{verb}, respectively, to a message containing them. We used T=10 for all the experiments as it is a reasonable upper limit on the number of named entities and verbs involved in an event. The constant k along with weights w_{cap} and w_{verb} were determined empirically by experimenting with many different values (section 3.2.1).

Given a message M of length N, its total weight W_M was computed using (1). Here, N was computed after discarding the stopwords and words of two characters or less. It works as a normalizing factor that penalizes too long messages if they don't contain top capitalized words or verbs.

$$\begin{aligned} W(M) = & \frac{k * |S_{\max}| + \sum_{i=1}^{N} W_i T_i}{N}, \\ & \text{where i } \in \{\text{cap, verb}\}\text{, and} \\ & T_i = \text{\# of unique words of type i in M} \end{aligned} \tag{1}$$

From the top weighted messages, M_{top}, a representative message was extracted as summary. For this, the centroid of M_{top} was computed and the message nearest to the centroid (in terms of cosine similarity) was selected as system summary.

As a preprocessing step, the outlier messages were removed by computing the centroid of the cluster of messages in bag of words representation, and discarding the messages that were too far from the centroid. Such messages don't represent the messages in the cluster and are likely to be noise messages. We discarded the farthest 20% messages for all the clusters. This step was performed for all summarizers. Another preprocessing step removed stopwords, words with less than 2 characters, and punctuations from the messages. During weight counting process, messages that contained excessive capitalization (capitalized words > 0.8 of total words) were also discarded as such messages are often non-event related but get higher w_{cap}. The proposed summariser (OurSum) is given in Algorithm 1.

Algorithm 1 OurSum

```
procedure SUMMARIZER(S)
    M ← Preprocess(S)
    Find_Top_Capitalized_Words_and_Verbs(M)
    Find_Sets_of_Frequent_Words(M)
    T ← Find_Top_Weighted_Messages(M) using (1)
    C ← Compute_Centroid(T)
    return Min(CosineDist(C, $T_i$)), for $T_i \in$ T
```

Comparing our summarizer with PR method of [9], the PR assigns more weight to a commonly used phrase. Our method generalizes that method to cover unordered words in a message. As an example, while the PR would consider a sequence of words $w_1w_2....w_n$ different from $w_2w_1...w_n$, our method treats them as same. This results in an advantage for our method as words of a message can be put in a valid but different order to create different phrases but with same meaning.

The sets of frequent words are analogous to frequent itemsets in market basket analysis. Each unique word of a preprocessed message can be considered as an item in the basket, each sentence as a transaction, and the cluster of messages as the basket of transactions. So we can apply a frequent itemset mining algorithm to find the set of the frequent common words. We used apriori algorithm [11] to find all frequent itemsets using support threshold of 10% (default in the tool used [12]). The support of a set of words in a cluster of messages is the percentage of messages in the cluster containing the set. So, each set of words that occurs in more than 10% of messages in a cluster is frequent.

There are a few parameters whose values need to be set. We now discuss the experiments performed to select these values.

3.2.1. Parameter analysis.

Multiple experiments with different proportions of weights for k, w_{cap} and w_{verb} were conducted, the results of a few of which are shown in Fig.1. In all the experiments, F-score (F1) in ROUGE-1 was used and our summarizer with different parameters was compared with a single set of model summaries that was labelled manually and subsequently extended using a method that is discussed in section 4.1.

The maximum score was obtained when k=2, w_{cap}=3, and w_{verb}=2. Increasing k further while keeping the other two parameters fixed didn't improve the result. A possible explanation for this behaviour is that the weight of the set of frequent words dominates the other two weights beyond a threshold. Giving weight of 0 to w_{cap} and w_{verb} limits the F-score to a maximum of 0.44 for all values of k. This is the maximum score the summarizer could score considering only the frequent set of words to assign weights (apart from the normalization and excessive capitalization removal). We

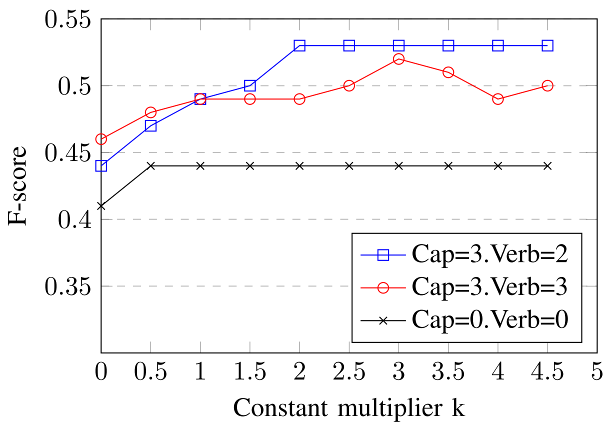


Figure 1. F-score for multiple combinations of k, weights of capital and verbs

finalised k=2 as the summarizer scored highest with k set to 2 in multiple different experiments. Keeping k fixed at 2, we ran our summarizer with many combinations of w_{cap} and w_{verb}, the results of a few of which are shown in Fig. 2. It can be seen that w_{cap}=3 with w_{verb}=2 maximized the score. We finalised these weights for our summarizer (OurSum).

4. Experiment

We used Twitter's streaming API to track generic English words, and used a bigram based approach of [13]. A total of 50 different clusters were classified as events and verified by manual inspection. These clusters were used for summarization task. The set of clusters contain a total of 3767 messages, with minimum and maximum cluster size of 16 and 275 messages, respectively.

4.1. Dataset of model summaries

We assigned the task of manual summary generation to three undergraduate students. They were given general instructions to write a summary for each of the 50 event clusters in their own way in less than 140 characters. Since different persons use different words to write summary of the same event, it gives rise to a significant limitation of metrics like ROUGE that counts overlapping words to compare system summary with manual summary. A message scores low in such metric if it does not contain the same words as in manual summary. Fig. 3a is such an example where two messages with the same meaning score differently in ROUGE. To keep this example simple, we have not removed stopwords.

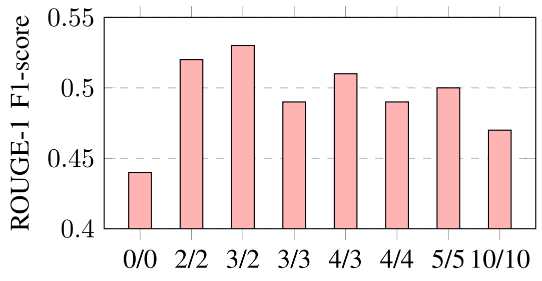
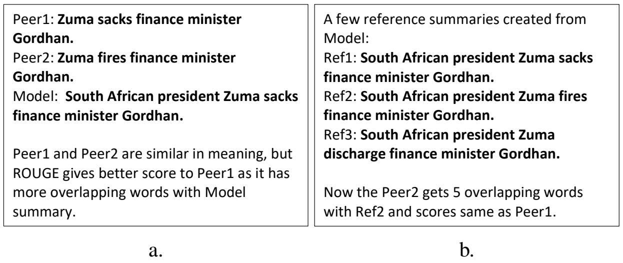


Figure 2. Performance of combinations of w_{cap}/w_{verb} with k fixed at 2.



Peer1: **Zuma sacks finance minister Gordhan.**
Peer2: **Zuma fires finance minister Gordhan.**
Model: **South African president Zuma sacks finance minister Gordhan.**

Peer1 and Peer2 are similar in meaning, but ROUGE gives better score to Peer1 as it has more overlapping words with Model summary.

a.

A few reference summaries created from Model:
Ref1: **South African president Zuma sacks finance minister Gordhan.**
Ref2: **South African president Zuma fires finance minister Gordhan.**
Ref3: **South African president Zuma discharge finance minister Gordhan.**

Now the Peer2 gets 5 overlapping words with Ref2 and scores same as Peer1.

b.

Figure 3. Overlapping words of system summary with a. single manual summary, and b. multiple reference summaries. For each peer summary, we consider the best score among all the reference summaries generated.

To handle this limitation, we extended each manual summary to multiple reference summaries by using synonyms for each word in the manual summary. We used synsets from Wordnet and a method similar to [10] but we only considered verbs for extension. Wordnet makes different meanings of a word available for different contexts called synsets. All words with similar meanings belong to a common synset. We can use various combinations of these synonyms to create many reference summaries for each manual summary. Instead of comparing to a single manual summary, the system summary is now compared to all the reference summaries generated for the manual summary, and the most overlapping reference summary's score is considered. To avoid generating unnecessarily big set of reference summaries, we limited the words in the following manner: We considered only single-word synonyms. We didn't consider capitalized words, stop words, and words with less than two characters. Since the context (called 'sense' in Wordnet) of a word in manual summary is not known, all contexts were assumed possible with equal probability. This would still create a large number of reference summaries for certain manual summaries, so we restricted the system to consider only the top ten synsets for each word, and for each synset the first five synonyms (lemmas). This generates a maximum of 50 words for a verb in manual summary. An example of a manual summary (Model) is shown in Fig. 3a and three of its reference summaries are shown in Fig. 3b. A better overlap of system summary (Peer2) with one of the

	Manual	Ref-1	Ref-2	Ref-3
Total Events	50	50	50	50
Total messages	50	73949	2103137	444106
Max file length	1	30720	1306368	274400
Min file length	1	3	6	4

TABLE 2. LENGTH OF REFERENCE SUMMARIES IN THE DATASET.

reference summaries can be seen in Fig. 3b as compared to Fig. 3a. Since the evaluation system uses the best performing reference summary among all the available ones, it helped ROUGE metrics to score system summaries more effectively as compared to evaluation with only a manual summary. The information on the size of reference summaries is given in Table 2.

All the summarizers were evaluated using ROUGE-N (N=1,2) and ROUGE-L metrics as they have been reported to have a good correlation with human written very short summary in Document Understanding Conferences (DUC)[14]. Overlapping n-grams between system summary and manual summary are used for measuring their similarity in ROUGE-N metric. In ROUGE-L, the longest common subsequence (LCS) is considered instead of n-grams. In ROUGE-N, the Precision (P) is computed as the ratio of the number of overlapping n-grams between system (peer) and model (reference) summaries to the number of n-grams in system summary, whereas Recall (R) is the ratio of the number of overlapping n-grams between system and model summaries to the number of n-grams in the model summary. In ROUGE-L, precision (P) is the ratio of length of the LCS of system and model summaries to the length of system summary, while recall (R) is the ratio of length of the LCS of peer and model summaries to the length of model summary. F1-score is the harmonic mean of P and R.

4.2. Results and Analysis

We compared each manual summary against the reference sets created from the other two summaries using ROUGE and the results are shown in Table 3. These results show us the expected accuracy scores, as manual summaries are supposed to be identical and informative. Still, there are many non-overlapping words in different manual summaries and their score never gets very high.

The summaries extracted by the summarizers (system summary) was evaluated against all the reference summaries for the corresponding clusters. To allow even more effective evaluation, three pairs of summaries were evaluated for each pair of system summary and reference summary. The first set contained the original summaries but stopwords were removed. Stemming was performed along with stopwords removal for creating the second set of summaries, and the third set was created by morphing the words of summaries after stopwords removal. Stemmer converts a word to its stem whereas morphing searches for a word form that is not in Wordnet. Since a cluster's system summary is used to create three sets of summaries, each message in a set of system summary is compared with all reference summaries in the similarly created set of reference summaries for that cluster. These steps help in getting more overlap among similar words.

	ROUGE-1			ROUGE-2			ROUGE-L		
	Man-1	Man-2	Man-3	Man-1	Man-2	Man-3	Man-1	Man-2	Man-3
Man-1	1	0.40	0.42	1	0.19	0.32	1	0.30	0.40
Man-2	0.38	1	0.40	0.17	1	0.19	0.30	1	0.34
Man-3	0.42	0.40	1	0.31	0.19	1	0.40	0.34	1

TABLE 3. F1-SCORES OF MANUAL SUMMARIES AGAINST EACH OTHER. THE MANUAL SUMMARIES IN EACH ROW WERE USED AS SYSTEM SUMMARY, AND EVALUATED AGAINST THE CORRESPONDING REFERENCE SUMMARIES FOR THE COLUMN.

	ROUGE-1			ROUGE-2			ROUGE-L		
	F1	P	R	F1	P	R	F1	P	R
Manual	0.40	0.45	0.38	0.23	0.26	0.22	0.35	0.39	0.33
Shortest	0.29	0.38	0.25	0.16	0.20	0.15	0.25	0.30	0.22
Longest	0.21	0.17	0.34	0.09	0.07	0.14	0.16	0.12	0.25
Random	0.28	0.25	0.35	0.10	0.10	0.14	0.22	0.20	0.29
Centroid	0.36	0.39	0.36	0.20	0.22	0.20	0.33	0.36	0.33
H-tfidf	0.42	0.45	0.41	**0.26**	**0.28**	**0.26**	0.38	0.41	0.37
OurSum	**0.45**	**0.47**	**0.45**	**0.26**	**0.28**	0.25	**0.40**	**0.42**	**0.39**

TABLE 4. AVERAGE F1-SCORES (F1) ON THE THREE REFERENCE SUMMARIES AND THE CORRESPONDING PRECISION(P) AND RECALL(R) USING ROUGE-1, ROUGE-2, ROUGE-L. ALL THE MAXIMUM SCORES ARE HIGHLIGHTED.

The F1-scores on the three sets of reference summaries for each event and the corresponding precision and recall of system summaries for all methods are given in Table 4. For row labelled Manual, the reported results are the average scores of the three manual summaries against the other extended reference sets. For summarizers, the best F1-scores of each system summary against every summary in a set of corresponding reference summaries was computed for all 50 event clusters and averaged. Since there were three sets of reference summaries corresponding to the three sets of manual summaries, we finally took the average of the best F1-scores on the three sets in Table 4 .

For the three sets of reference summaries, the macro-averaged F_{avg} of these three sets, and the corresponding precision and recall scores are shown. Since manual summaries Man-1 and Man-3 were the most similar among the three, we also report the average ROUGE-1 score of the summarizers on these two sets of summaries in Fig. 4 using the best parameters that maximized the scores of H-tfidf and OurSum.

We used F1-score instead of only recall or precision as it provides a good balance between the two. We can be reasonably sure of a good precision and recall if the F1-score is good. As extreme examples, a good precision and bad recall can be achieved in ROUGE by generating a small summary containing words that are most common in the messages. Conversely, a good recall but bad precision can be achieved by creating a long summary by union of words from multiple messages. F1-score puts a check on such cases by leaning more towards the lower value among precision and recall.

In our experiments, Shortest summarizer performed better than Longest and Random summarizers in all the ROUGE scores. It seems that, on average, the shorter tweets are more meaningful than the longer ones. The Longest

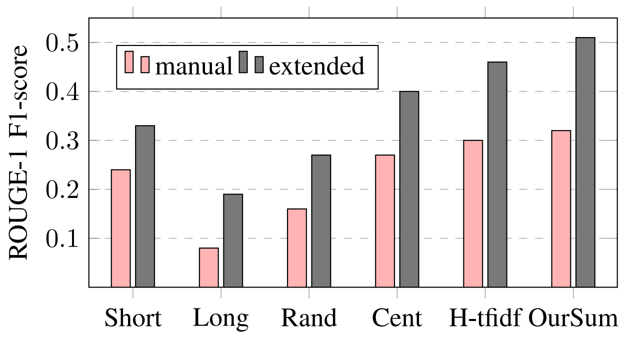


Figure 4. Comparison of performance of all summarizers on the two high scoring sets of manual and extended summaries.

summarizer was particularly weak in ROUGE-2 scores. On closer inspection of the manual summaries, our summarizer seems to favour shorter summaries while Hbrid-tfidf favoured longer ones. While the two of the manual summaries were shorter with average of 8.4 words per message, the third one was longer with 11.6 words per message, and Hybrid-tfidf performed its best on that summary, sometimes better than our summarizer. Our summarizer performed much better on the two shorter summaries for an overall good score. The performance of Hybrid-tfidf was comparable to OurSum in ROUGE-2, but OurSum scored better in ROUGE-1 and ROUGE-L. We can modify the normalization factor to give preference to shorter or longer system summaries, but we leave that as a future work.

OurSum effectively used frequent itemset mining to assign weights to commonly occurring sets of words in a cluster of related messages. Also, the results show that despite being unstructured and not following grammatical rules when tweets are viewed individually, they collectively tend to offset this effect. So, we were able to assign weights to messages containing the most commonly used capitalized and action words to create an effective summarizer.

5. Conclusion

In this paper, we used frequent sets of words along with capitalization and verb based weight assignments to create a summarizer (OurSum). It summarizes clusters of related messages by extracting the most representative message that is concise and provides most information about the event. OurSum uses centroid property to remove outliers and to extract the highest weighted message as system summary. Assignment of weights to messages was performed partially by applying a frequent itemset mining algorithm. We compared our proposed summarizer on standard ROUGE based metrics with many baseline methods. We created extensive reference summaries from a small set of manual summaries using Wordnet. On average, our summarizer performed well in the experiments outperforming others summarizers. As a future work, the dataset size could be further increased to contain more event clusters. More manual summaries could be added to arrive at a conclusion about the general behaviour of the various parameters. More fine-grained weights could be included by using linguistic properties. Another interesting work would be extract important information such as time or duration of event, key entities, and action involved in an event to present to a user.

References

[1] Sakaki, T., Okazaki, M., Matsuo, Y., ”Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors”, Proceedings of the International conference on World Wide Web, Raleigh, North Carolina, USA, 2010, Apr 26-30, 851-860.

[2] Becker, H., Naaman, M., Gravano, L., Beyond Trending Topics: Real-World Event Identification on Twitter”, Proceedings of the International AAAI Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, 2011, July 17-21, 438-441.

[3] Miller, G.A., ”WordNet: A Lexical Database for English”, Communications of the ACM Vol. 38, No. 11: 39-41 1995.

[4] Ganesan, K., Zhai, C., and Viegas, E., ”Micropinion generation: an unsupervised approach to generating ultra-concise summaries of opinions”. In: Proceedings of the International World Wide Web Conference, Lyon, France, April 16-20, 2012.

[5] Chakrabarti, D., Punera, K.: ”Event Summarization using Tweets”, Proceedings of the International Conference on Web and Social Media, Barcelona, Spain, July 17 21, 2011.

[6] Nichols, J., Mahmud, J., Drews, C., ”Summarizing Sporting Events Using Twitter”, Proceedings of the International conference on Intelligent User Interfaces, Lisbon, Portugal, February 1417, 2012.

[7] Chua, F.C.T., Asur, S., ”Automatic Summarization of Events From Social Media”, Seventh International AAAI Conference on Weblogs and Social Media, Cambridge, Massachusetts, USA, July 811, 2013.

[8] Baralis, E., Cagliero, L., Jabeen, S., Friori, A., ”Multi-document summarization exploiting frequent itemsets”, Proceedings of the 27th Annual ACM Symposium on Applied Computing, Trento, Italy, March 26-30, 2012.

[9] Sharifi, B., Hutton M.A., Kalita, J., ”Experiments in Microblog Summarization”, Proceedings of the International Conference on Social Computing (SocialCom), Minneapolis, MN, USA, August 20-22, 2010.

[10] Samant, S.S., Bhanu Murthy, N.L., Malapati, A., ”CapAct: A Wordnet-based Summarizer for Real-World Events from Microblogs”, Proceedings of the International Conference on Distributed Computing and Internet Technology (ICDCIT), Bhubaneswar, India, January 11-13, 2018. In press.

[11] Agrawal, R., Srikant, R., ”Fast algorithms for mining association rules”, Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487-499, Santiago, Chile, September 1994.

[12] Borgelt, C., ”Efficient Implementations of Apriori and Eclat”, Workshop of Frequent Item Set Mining Implementations (FIMI) 2003.

[13] Samant, S.S., Bhanu Murthy, N.L., Malapati, A., ”Bigram-based Features for Real-World Event Identification from Microblogs”, Proceedings of the International Conference on Computing, Communication and Networking Technologies (ICCCNT), New Delhi, India, July 3-5 2017.

[14] Lin, L.Y., ”Looking for a Few Good Metrics: ROUGE and its Evaluation”, Working Notes of NTCIR-4, Tokyo, June 2-4, 2004.