SJIF Impact Factor 2022: 8.197| ISI I.F. Value:1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)

EPRA International Journal of Research and Development (IJRD)

Volume: 7 | Issue: 5 | May 2022 - Peer Reviewed Journal

IMPLEMENTING THE EXTRACTIVE DISASTER TWEET SUMMARIZATION ALGORITHMS AND ANALYZING THEIR EFFICIENCY

Karuna Middha[1], Harshit Garg[2], Kshem Sharma[3], Rohan Aggarwal[4]

[1]Maharaja Agrasen Institute of Technology (Assistant Professor) GGSIPU,Delhi, India
[2]Maharaja Agrasen Institute of Technology (CSE) GGSIPU,Delhi, India
[3]Maharaja Agrasen Institute of Technology (CSE) GGSIPU,Delhi, India
[4]Maharaja Agrasen Institute of Technology (CSE) GGSIPU,Delhi, India

ABSTRACT

Social Media Websites, predominantly Twitter have become important sources for real-time situational information during emergency events like the entire country witnessed during the second wave of Covid-19 in our country. People turned to each other for help to arrange for beds, oxygen concentrators, etc. Since hundreds to thousands of microblogs or tweets are generally posted on Twitter during an emergency event, manually going through every tweet is not feasible. In such a scenario, it is critical to summarize the microblogs (tweets) and present informative summaries to the people who are attempting to respond to the disaster.

KEYWORDS — *Twitter, Extractive Summarization, Comparison Evaluation, Rouge*

I. INTRODUCTION

Microblogging services such as Twitter have become incredibly essential sources of real-time information about ongoing events such as sociopolitical events, natural and man-made disasters, and so on. Microblogging sites are critical sources of situational information, especially during emergency situations such as disasters. During such situations, microblogs are generally uploaded so quickly and in such enormous volumes that human users are unable to read them all. In such a case, it is vital to summarize the microblogs (tweets) and offer helpful summaries to those attempting to respond to the tragedy.

Automatic document summarization[1] is a well-established problem in Information Retrieval, and many algorithms have been proposed for the problem. Summarization methods are broadly of two types—abstractive and extractive. While extractive algorithms generate summaries by extracting certain portions of the input data (e.g., certain sentences that are deemed important), abstractive algorithms attempt to generate summaries by paraphrasing parts of the input data. Out of these, most of the algorithms proposed in literature are extractive in nature.

With the growing prominence of microblogs as a source of information, a variety of summarising algorithms for microblogs have recently been presented. The difficulty of summarising microblogs is fundamentally a multi-document summary problem. Algorithms for single-document summarization, which consider the input set of microblogs to form a single document, are also relevant. Microblog summarization presents several unique issues, owing to the short size of individual microblogs and the loud, informal style of microblogs, which makes interpreting semantic similarity challenging.

There are several summarising methods in the literature, both for general documents and particularly for microblogs.

However, to the best of our knowledge, no systematic research has been conducted to determine how useful these algorithms are in the application of summarising microblogs generated during crisis occurrences. In this paper, we assess and compare eight commercially available extractive summarization techniques for the aforementioned application. We conduct tests on microblogs on five recent catastrophic incidents. We find that several off-the-shelf algorithms produce radically different summaries from the same set of microblogs, with relatively few tweets shared by the summaries produced by different algorithms. Furthermore, we use the usual ROUGE measurement to assess the performance of the various methods. In comparison to the other algorithms covered here, the LUHN and MEAD algorithms produce quite high ROUGE ratings.

II. PROPOSED ARCHITECTURE

We detail the extractive summarization methods that we considered for comparison in the current study in this section. It should be noted that several of these algorithms were originally developed for summaries of a single document, in which the sentences of the provided text are sorted according to some importance measure, and then a few key lines are chosen for the summary. These algorithms are easily applicable to summarising a collection of microblogs, where each microblog equates to a sentence.

(1) **Cluster Rank:**ClusterRank is an unsupervised, graph-based technique that was developed originally for extractive summarization of meeting transcripts. The Cluster-Rank algorithm is an extension of the TextRank algorithm, which is likewise a graph-based approach for extracting sentences from news articles. ClusterRank divides the transcript into clusters, which are represented as graph nodes. The similarity between all nearby cluster pairs is then calculated, and the pair with the highest similarity is combined into a single cluster. Following that, each sentence inside an important cluster is scored using a centroid-based technique. In addition to dealing with ill-formed phrases with significant repetition, the relevance of the sentences is also examined. Finally, the algorithm chooses the highest scoring sentence and inserts it in the summary until the length restriction is met.

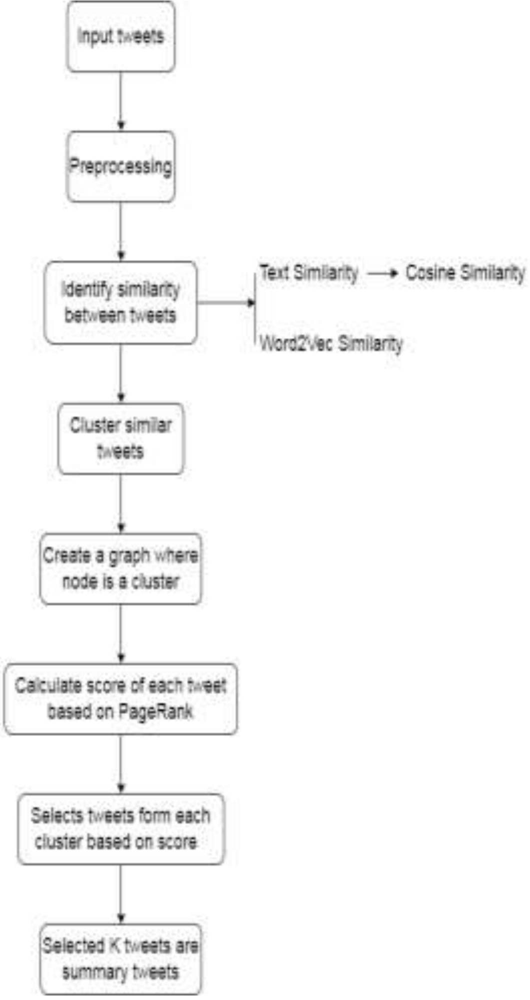

Input tweets
Preprocessing
Identify similarity between tweets
Text Similarity
Cosine Similarity
Word2Vec Similarity
Cluster similar tweets
Create a graph where node is a cluster
Calculate score of each tweet based on PageRank
Selects tweets form each cluster based on score
Selected K tweets are summary tweets

Fig 1.1: Cluster Rank

(2) **Mead:** Mead is a centroid-based multi-document summarizer. First, subjects are identified via agglomerative clustering on the papers' tf-idf vector representations. Second, a centroid-based technique is employed to find phrases in each cluster that are essential to the overall cluster's issue. Three separate properties are computed for each sentence: centroid value, positional value, and first-sentence overlap. The three ratings are used to get a composite score for each sentence. The score is adjusted further after taking into account probable cross-sentence relationships, such as repeated sentences, chronological ordering, and source preferences.) Finally, sentences are chosen depending on this. score.

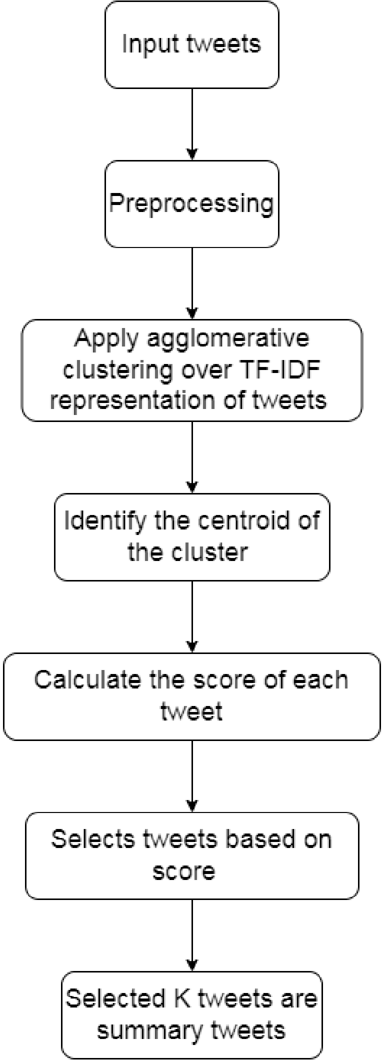

Input tweets
Preprocessing
Apply agglomerative clustering over TF-IDF representation of tweets
Identify the centroid of the cluster
Calculate the score of each tweet
Selects tweets based on score
Selected K tweets are summary tweets

Fig 1.2: MEAD

(3) **DepSub**: DEPSUB[10], or "Dependency-Parser-based SUB-event detection," detected noun-verb pairings indicating sub-topics — such as "bridge collapse" or "person imprisoned" — and rated them based on how frequently they appear in tweets.

Then it creates summaries of the overall event as well as the detected sub-events.

The parser builds a dependency tree for each tweet iteratively, removing any parent and child nodes that are nouns or verbs. This approach identifies a huge number of noun-verb pairings, and many of the candidates are not sub-events. We only evaluate noun-verb pairings that appear more than once in the dataset to filter out the noisy pairs. Consider the following processed tweet: waterborne distress eases as hurricane water recedes. "Waterborne recedes" and "water recedes" are identified as noun-verb pairs by the dependency parser. "Waterborne diseases" is recognised as a phrase by the phrase model. Their composition is thought of as candidate sub-events.

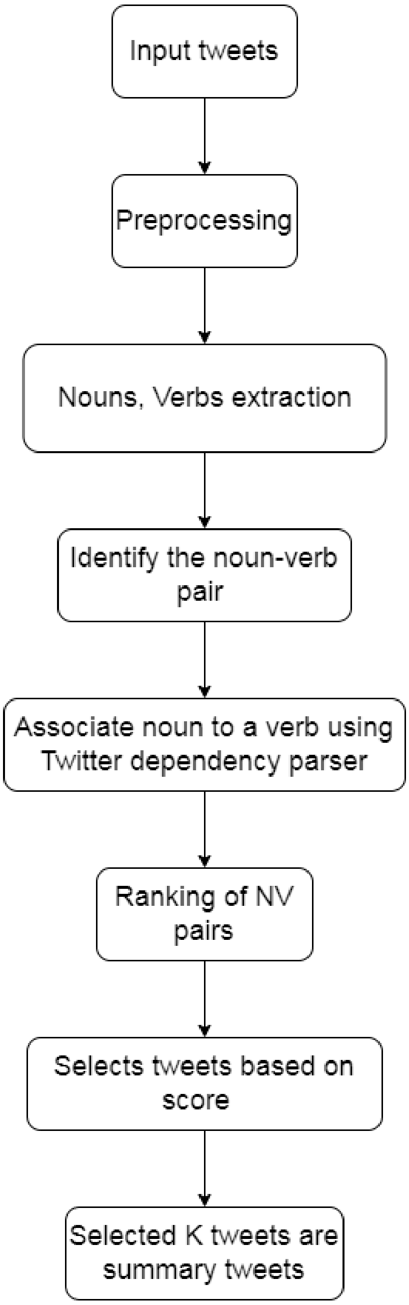

Input tweets
Preprocessing
Nouns, Verbs extraction
Identify the noun-verb pair
Associate noun to a verb using Twitter dependency parser
Ranking of NV pairs
Selects tweets based on score
Selected K tweets are summary tweets

Fig 1.3: DepSub Algorithm

(4) **LUHN:** Luhn's technique[2] is based on the impression that some words in a text are descriptive of its content, and the sentences that represent the most important information in the document include a high density of such descriptive terms. Words that appear often in a text are likely to be associated with the document's main theme. Stopwords, on the other hand, are an exception to this rule. As a result, Luhn recommended the use of stopwords like as determiners, prepositions, and pronouns, which have little value in telling the reader about the topic of the document. As a result, he proposed deleting these terms from consideration. Luhn used

experimentally derived high- and low-frequency criteria to identify descriptive terms.

High-frequency thresholds exclude terms that appear often throughout the content. Similarly, low-frequency thresholds exclude words that appear seldom. The remaining words in the paper are descriptive terms that indicate the significant material.

A 'significance factor' is computed for each phrase, which may be derived by bracketing the significant words in the sentence, squaring the number of significant words, and then dividing by the total number of words. Based on the importance factor values, sentences are determined as important and included in the summary.

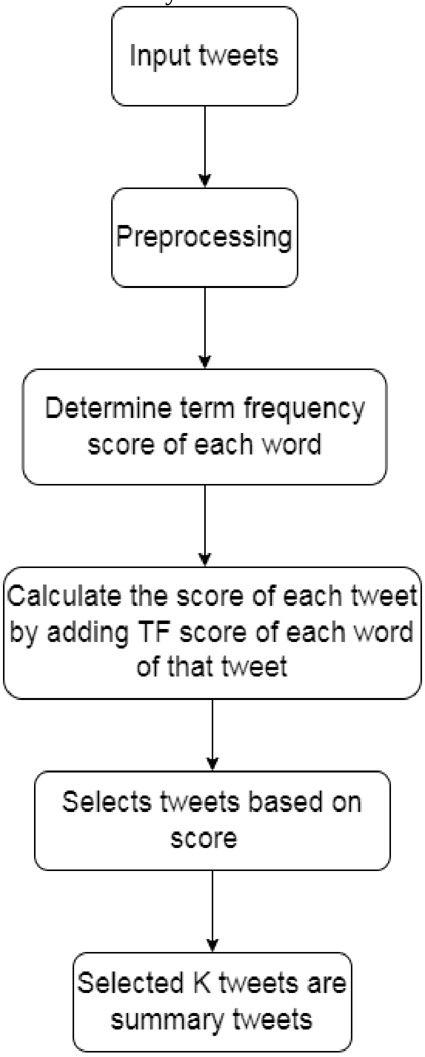

Input tweets
Preprocessing
Determine term frequency score of each word
Calculate the score of each tweet by adding TF score of each word of that tweet
Selects tweets based on score
Selected K tweets are summary tweets

Fig. 1.4: LUHN Algorithm

(5) **SumBasic:** SumBasic is a multi-document summarizer based on frequency. SumBasic computes the probability distribution across the words of a phrase using a multinomial distribution function. Scores are provided to each sentence based on the average frequency of occurrence of the terms in the sentence. The sentences with the highest ratings are then chosen. The word probabilities and sentence scores are incrementally updated until the required summary length is attained. Updating word probabilities is a logical technique of dealing with duplication in multi-document input.

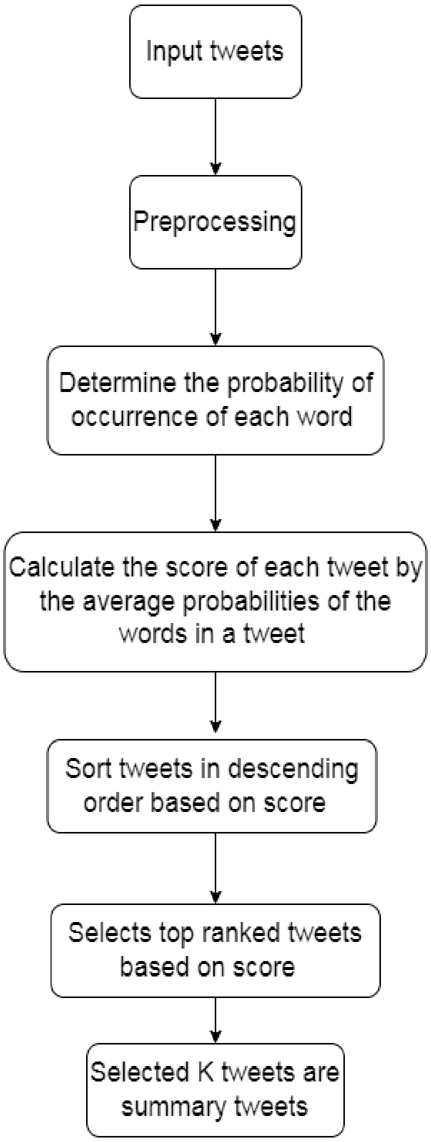

Input tweets
Preprocessing
Determine the probability of occurrence of each word
Calculate the score of each tweet by the average probabilities of the words in a tweet
Sort tweets in descending order based on score
Selects top ranked tweets based on score
Selected K tweets are summary tweets

Fig 1.5: SumBasic Algorithm

The dataset that was utilised is as follows:

1. SH_Shoot(D1) – A shooter killed 20 children and six adults at Sandy Hook Elementary School in Connecticut, USA.
2. U_Flood(D2) - severe floods and landslides in India's Uttaranchal state
3. T_Hagupit(D3) – Typhoon Hagupit, a powerful storm, struck the Philippines.

III. RESULT

To assess the quality of an algorithm-generated summary, we follow the conventional approach of creating gold standard summaries by human annotators and then comparing the algorithm-generated summary to the gold standard summaries generated by us.

On each dataset, we ran all of the summarizing methods and obtained summaries of 25 tweets in length. We utilized the standard ROUGE metric to assess the quality of the summaries generated by different algorithms based on their similarity to the gold standard summaries. Due to the informal character of tweets, we investigated the Recall and F-score of the ROUGE-1, ROUGE-2, and ROUGE-L variations.

The results obtained on all three datasets are as follows:

Approach	D1		
	ROUGE-1	ROUGE-2	ROUGE-L
ClusterRank	0.46	0.19	0.25
MEAD	0.41	0.16	0.21
DepSub	0.48	0.20	0.27
LUHN	0.47	0.19	0.26
SumBasic	0.44	0.17	0.23

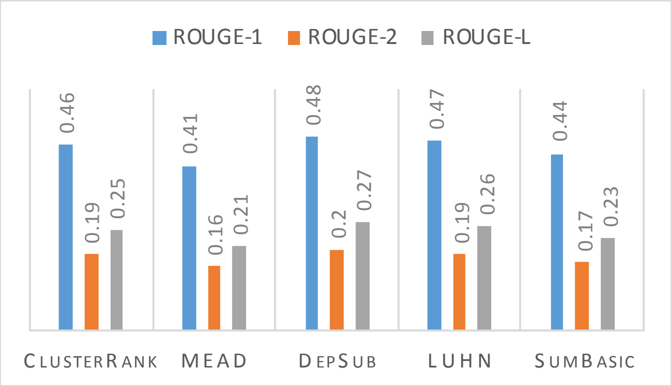
Table 1: Performance of Summarization Algorithms on D1 Dataset

Approach	D2		
	ROUGE-1	ROUGE-2	ROUGE-L
ClusterRank	0.39	0.11	0.23
MEAD	0.32	0.07	0.19
DepSub	0.41	0.12	0.24
LUHN	0.34	0.09	0.20
SumBasic	0.37	0.10	0.21

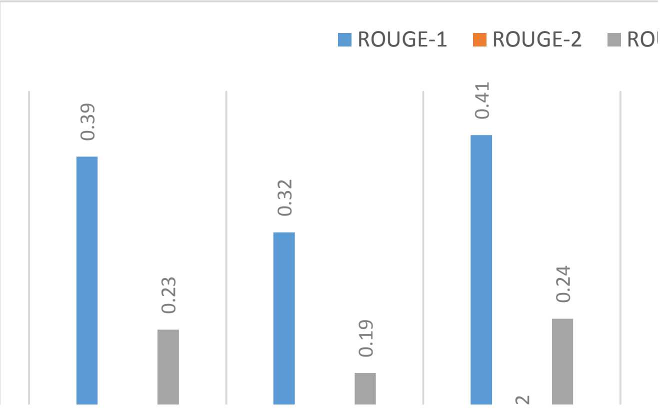
Table 2: Performance of Summarization Algorithms on D2 Dataset

Approach	D3		
	ROUGE-1	ROUGE-2	ROUGE-L
ClusterRank	0.41	0.16	0.20
MEAD	0.43	0.15	0.22
DepSub	0.45	0.17	0.24
LUHN	0.42	0.15	0.19
SumBasic	0.41	0.16	0.20

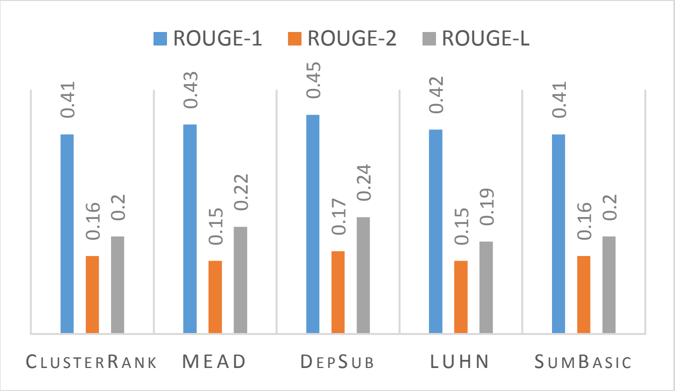
Table 3: Performance of Summarization Algorithms on D3 Dataset



Graph 1: Performance of Summarization Algorithms on D1 Dataset



Graph 2: Performance of Summarization Algorithms on D2 Dataset



Graph 3: Performance of Summarization Algorithms on D3 Dataset

IV. CONSLUSION

A significant and practical issue is the summarization of microblogs made during emergency circumstances. While several summary algorithms have been presented in the literature, to our knowledge, no systematic comparison of how successful different algorithms are at summarising microblogs connected to disaster occurrences has been conducted. In this paper, we compare five extractive summarization algorithms on microblogs produced after three crisis incidents. We discover that different algorithms provide dramatically varied summaries, and that while some algorithms (e.g., DepSub) attain reasonably high ROUGE scores, others, such as MEAD, do not appear to be as successful.

We believe that the current study points to various potential research avenues. First, considering that even the top performing approaches attain ROUGE recall scores of less than 0.6, improved algorithms are clearly required for properly summarising microblogs during crisis occurrences. Second, because various summarization algorithms create quite different summaries from the same input data, it may be worthwhile to study if the outputs of numerous summarization methods may be merged to produce summaries that outperform the individual techniques.

V. REFERENCES

1. *Moratanch, N., Gopalan Chitrakala A survey on extractive text summarization. 2017*
2. *Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev.* ***2****(2), 159–165 (1958)*
3. *Olariu, A.: Efficient online summarization of microblogging streams. In: Proceedings of EACL(short paper), pp. 236–240 (2014)*
4. *Gan G, Ma C, Wu J. Data clustering: theory, algorithms, and applications. vol. 20. Siam; 2007.*
5. *Zhang T, Ramakrishnan R, Livny M. BIRCH: an efficient data clustering method for very large databases. ACM Sigmod Record. 1996*
6. *Rosa,K.D.,Shah,R.,Lin,B.,Gershman,A.,Frederking,R.:Topical Clustering of Tweets*
7. *Das, D., Martins, A.F.: A survey on automatic text summarization. Lit. Surv. Lang. Stat. II Course CMU* ***4****, 192–195 (2007)*
8. *Gupta,V.,Lehal,G.S.:A survey of text summarization extractive techniques.IEEEJ.Emerg. Technol. Web Intell.* ***2****(3), 258–268 (2010)*
9. *Imran,M.,Castillo,C.,Diaz,F.,Vieweg,S.:Processing social media messages in massemergency: a survey. ACM Comput. Surv.* ***47****(4), 67:1–67:38 (2015)*
10. *Banerjee, Subhankar & Chakraborty, Shayok. (2019). Deepsub: A Novel Subset Selection Framework for Training Deep Learning Architectures. 1615-1619. 10.1109/ICIP.2019.8803096.*