A Survey on Template Based Abstractive Summarization of Twitter Topic Using Ensemble SVM with Speech Act .

Gulab R. Shaikh, Digambar M. Padulkar
ME(Computer), Assistant professor at VPCoE Baramati.

Abstract

In this todays Internet world, social media service such as Twitter is very popular and used worldwide. In this survey paper we are going to describe various methodology and approches that are useful in twitter topic summarization. In this survey paper, we have described various approch such as analysing and summarizing of twitter tweets using SPUR (Summarization via Pattern Utility and Ranking). In Participant-based Event Summarization we are able to identify each participant that are included in event and generate summary of that event. In automatic summarization of social media we have described a search and summarization framework to extract relevant representative tweets from an unfiltered tweet stream. Also we have described sequential summarization technique for summary creation that uses stream-based approach and the semantic-based approach that create good quality summary. We have also described Phrase Reinforcement algorithm and PageRank algorithm that generate summary by extracting information from the tweets.

Twitter topic summarization must handle various numerious, short, dissimilar and noisy nature of tweet. So all the approches that we have described here have its own advantages and limitation over the other.

In this survey paper we have try to describe various implemented approches, algorithm, evalution technique and their advantages and limitations.

1. Introduction

In the age of social media, Twitter is very popular social networking site. According to a Wall Street Journal report the microblogging service of Twitter spews out over 200 million tweets every day. The top trending topics on **Twitter.com** each can have thousands of tweets or more, It is very time consuming and deterring attempts to read all the tweets under a topic. Summarizing Twitter topics, however, is a very different challenge from summarizing other genres of text such as news articles, research papers, books, etc. Tweets contain a wide variety of useful information from many perspectives about important events taking place in the world. The huge number of messages, many containing irrelevant and redundant information, quickly leads to a situation of information overload. This motivates the need for automatic summarization systems which can select a few messages for presentation to a user which cover the most important information relating to the event without redundancy and filter out irrelevant and personal information.

Twitter is highly attractive for information extraction and text mining purposes, because they offer large volumes of real-time data. The quality of messages varies significantly, however, ranging from high quality text to meaningless strings. Typos, ad hoc abbreviations, phonetic substitutions, ungrammatical structures and emoticons etc.

Microblogging, a lightweight and easy form of communication within social networks such as Facebook, Google+and Twitter, has become ubiquitous in its use with over 4 billion mobile devices worldwide of which over 1 billion support smart services. An increasing number of organizations and agencies are turning to extract and analyze useful nuggets of information from such services to aid in functions as diverse as emergency response, viral marketing, disease outbreaks, and predicting movie box office success.

1. A Framework for Summarizing and Analyzing Twitter Feeds

1.1.Introduction

In this paper [15], author present a dynamic pattern driven approach to summarize data produced by Twitter feeds. Here they have develop a novel approach to maintain an in-memory summary while retaining sufficient information to facilitate a range of userspecific and topic-specific temporal analytics.

One of the most difficult task in social networking is that answering the complex query. For example query like: What topics were people talking about in a specific time interval .? How has a particular topic evolved across multiple time intervals .? How have a user's or a group's tweets, or topics they tweet on changed over time.? Answering such queries in real-time is challenging simply because of the scale of the data that is produced .

In this work [15], the main aim is to build a summary of microblogging data, focusing on Twitter feeds, that can fit in a limited memory budget and can help to answer complex queries. The elements of framework include:

a) SPUR, a batch summarization and compression algorithm that relies on a novel notion of pattern utility and ranking which can be incrementally updated.

b) D-SPUR, a dynamic variant of SPUR that accordingly merges summaries and maintains pyramidal time frames that grows logarithmically while enabling querying at multiple temporal contexts.

c) TED-SPUR, a topic and event based analytics tool to support complex querying on dynamic data.

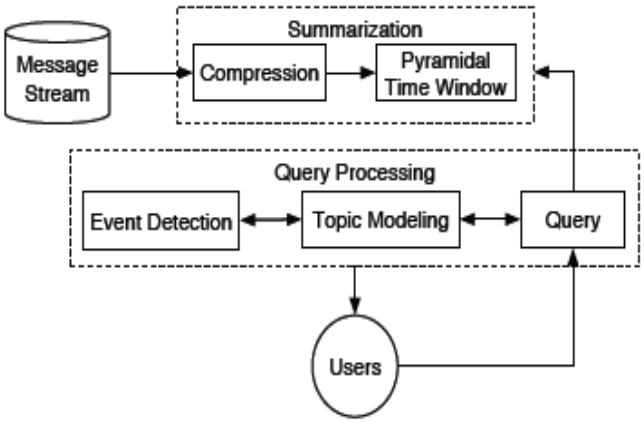

Figure 1: SPUR framework

1.2.Stream Summarization

In this topic [15], authors have introduced the summarization component of stream processing framework.

Figure 2: Division and compression

Given the input message stream with proper word stemming and stop-word removal performed, they divide it into approximately equal-sized batches, e.g. one hour per batch (the first arrow in Figure 2). To compress each batch of messages into a summary object which can fit in a constant memory budget M .

1.2.1. SPUR

Here, in this work [15] author developed a novel algorithm called SPUR (Summarization via Pattern Utility and Ranking) to summarize a batch of transactions with low compression ratio and high quality in a highly scalable fashion. The basic idea of compressing a batch of tweets is to replace individual words with frequently used phrases that can cover the original content of the tweets.

1.2.2. D-SPUR

D-SPUR [15], it is the dynamic version of SPUR. In D-SPUR, is used to enhance and modify the pyramidal time window for clustering in data streams. The main objective is to find an effective way to manage the stream of summary objects produced by SPUR while limiting the growth of memory footprint and reconstruction error (especially on recent and trending data). D-SPUR summarizes dynamic message streams by maintaining the pyramidal time window .Using the

pyramidal time window, the total memory footprint will grow logarithmically. More importantly, historical data is more compressed than recent data, so the summary is more accurate in recent time intervals.

1.2.3. Live Analytics with TED-SPUR

Lastly, Author present the query processing component of this framework, an analytical tool TED-SPUR (Topic and Event detection with D-SPUR), to support complex queries on dynamic data.

1.3.Advantages and Limitations

1.Stream Summarization framework, which can incrementally build summaries of Twitter message streams with one-pass over the data.

2.It compress twitter messages with low compression ratio, high quality and fast running time.

3.The memory footprint of stream summarization algorithm grows approximately logarithmically with time.

4.When comparing this approach with several state-of-the-art pattern summarization approaches along the axes of storage cost, query accuracy, query flexibility, and efficiency using real data from Twitter. Author find that this approach is not only scalable but also outperforms existing approaches by a large margin.

There are three main challenges one need to address: compressability, scalability and quality (of compressed summary).

2. A Participant-based Approach for Event Summarization Using Twitter Streams

2.1.Introduction

Event summarization mainly focusing on developing accurate sub-event detection systems and generating text descriptions that can best summarize the sub-events in a progressive manner. Here [17], authors described novel participant based event summarization approach, which dynamically identifies the participants from data streams, then "zooms-in" the event stream to participant level, detects the important sub-events related to each participant using a novel time-content mixture model, and generates the event summary progressively by concatenating the descriptions of the important sub-events.

2.2 Participant-based Event Summarization

Participant-centered event summarization approach consists of three key components:

2.2.1. Participant Detection

It dynamically identifies the event participants and divides the entire event stream into a number of participant streams.

2.2.2.Sub-event Detection

Here, authors have introduced a novel time-content mixture model approach to identify the important sub-events associated with each participant.These "participant-level sub-events" are then merged along the timeline to form a set of "global sub-events" , which capture all the important moments in the event stream.

2.2.3.Summary Tweet Extraction

Finally,extracts the representative tweets from the global sub-events and forms a comprehensive coverage of the event progress .

2.3. Advantages and Limitattions

Previous research on event summarization focuses on identifying the important moments from the coarse-level event stream. This may yield several side effects:

1.The spike patterns are not clearly identifiable from the overall event stream, though they are more clearly seen if we "zoom-in" to the participant level.

2.It is arguable whether the important sub-events can be accurately detected based solely on the tweet volume change.

3. A popular participant or sub-event can elicit huge volume of tweets which dominant the event discussion and shield less prominent sub-events.

For example, in the IPL Cricket game, discussions about the key players (e.g., “MS Dhoni”, “Sachin Tendulkar”) can heavily shadow other important participants or sub-events, resulting in an event summary with repetitive descriptions about the dominant players.

3.Automatic Summarization of Events from Social Media

Social media services such as Twitter generate large volume of content for most real-world events on a daily basis. Digging through the noise and redundancy to understand the important aspects of the content is a very challenging task. Here , author described a search and summarizationframework to extract relevant representative tweets from an unfiltered tweet stream in order to generate a coherent and concise summary of an event.

3.1. Summarization Framework

To summarize for the event of interest e

from the unfiltered tweet stream D, first assume that a set of queries Q, where each query $q \in Q$ is defined by a set of keywords. For example, the set of queries for the event “Facebook IPO” can be { { facebook, ipo }, { fb, ipo }, { facebook, initial, public, offer }, { fb,initial, public, offer }, { facebook, initial, public, offering }, { fb, initial, public, offering } }.

1. From D, extract all tweets that match at least one of the queries $q \in Q$. A tweet matches a query q if it contains all of the keywords in q. The set of tweets 1 obtained is denoted by D_e^1.

2. Next, apply a topic model on , D_e^1 to find keywords that describe the main aspects of the event that are being discussed. Then author developed two topic models that are designed to extract relevant tweets.

3. Once they have obtained the set of topics Z from the topic models, the top ranked words in each topic $z \in Z$ are the keywords that describe various aspects of the event e. Author may obtain the additional set of tweets D_e^2 by finding tweets $d \in D$ that are not present in D_e^1 by selecting those with high perplexity score with respect to the topics.

4. D_e^1 and D_e^2 can be merged to refine the model and improve upon the topics for the event e.

5. Using the final set of topics $z \in Z$, able to summarize the event e by selecting the tweets d from each topic z that give the best (lowest) perplexity.

The whole process can be performed for several iterations to improve the quality of the summary.

3.2.Comparision with Micro-Blog Event Summarization

Chakrabarti and Punera have proposed a variant of Hidden Markov Models to obtain an intermediate representation for a sequence of tweets relevant for an event . Their approach does not use the continuous time stamps present in tweets and does not address the problem of obtaining the minimal set of tweets relevant to an event.

Meng et al. have summarized opinions for entities in Twitter by mining hash-tags to infer the presence of entities and inferring sentiments from tweets. However, not all tweets contain hash-tags which makes it difficult to gain sufficient coverage for an event this way.

Sharifi et al. have proposed the Phrase Reinforcement Algorithm to find the best tweet that matches a given phrase, such as trending keywords. They produce one tweet as a summary for one phrase while we propose to provide a set of tweets to summarize an event.

Yang et al. have also proposed a framework for summarizing the unfiltered tweet stream. Their main focus is on creating a scalable approach by compressing the tweet stream to fit in limited memory, followed by the use of Non-negative Matrix Factorization (NNMF) to find topics in the tweet stream. Since they do not filter the tweets for a specific event of interest, the topics discovered using their framework will only contain globally major events. This proposed framework finds a summary for a targeted event of interest.

4.LexRank: Graph-based Centrality as Salience In Text Summarization

Here [7], author describe concept of graph based lexical ce ntrality for the text summarization. Extractive Text Summarization based on the concept of sentence salience to identify the most important sentences in a document or set of documents. Salience is typically defined in terms of the presence of particular important words or in terms of similarity to a centroid pseudo-sentence.

Here, in this work [7] they described a new technique called LexRank. It is used for computing sentence importance based on the concept of eigenvector centrality in a graph representation of sentences.

In this paper, author mainly focus on multi-document extractive generic text summarization, where the goal is to produce a summary of multiple documents about the same, but unspecified topic.

4.1. Extractive and Abstractive Summarization

Extractive summarization produces summaries by choosing a subset of the sentences in the original document. This contrasts with abstractive summarization, where the information in the text is rephrased. Now a days , most of the summarization research is on extractive summarization.

4.2. Sentence Centrality and Centroid-based Summarization

A common way of assessing word centrality is to look at the centroid of the document cluster in a vector space. The centroid of a cluster is a pseudo-document which consists of words that have tf×idf scores above a predefined threshold, where tf is the frequency of a word in the cluster, and idf values are typically computed over a much larger and similar genre data set. In centroid-based summarization the sentences that contain more words from the centroid of the cluster are considered as central .

One of the main problems with multi-document summarization is the potential duplicate information coming from different documents. In such case, author try to avoid the repeated information in the summaries by using the reranker of the MEAD system. However , Instead of using a reranker, they first segment the text into regions of different subtopics and then take at least one representative paragraph with the highest degree value from each region.

To determine the similarity between two sentences, they have used the cosine similarity metric that is based on word overlap and idf weighting. Also, the similarity computation might be improved by incorporating more features (e.g. synonym overlap,verb/argument structure overlap, stem overlap) or mechanisms into the system.

4.3. Advantages and Limitations

In this work, authors have presented a new approach to define sentence salience based on graph-based centrality scoring of sentences. Constructing the similarity graph of sentences provides us with a better view of important sentences compared to the centroid approach, which is prone to over-generalization of the information in a document cluster.

In LexRank, authors have tried to make use of more of the information in the graph, and got even better results in most of the cases. Lastly, they have shown that this methods are quite insensitive to noisy data that often occurs as a result of imperfect topical document clustering algorithms.

5. TextRank: Bringing order into texts

Graph-based ranking algorithms [2] are very useful in deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. The basic idea implemented by a graph-based ranking model is that of voting . When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex.

5.1. Text as a Graph

The main task here is that, build a graph that represents the text, and interconnects words or other text entities with meaningful relations. Graph based ranking algorithms consists of the following main steps:

1. Identify text units that best define the task at hand, and add them as vertex in the graph.

2. Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph. Edges can be directed or undirected, weighted or unweighted.

3. Iterate the graph-based ranking algorithm until convergence.

4. Sort vertices based on their final score. Use the values attached to each vertex for ranking/selection decisions.

5.2. TextRank for Keyword Extraction

The TextRank keyword extraction algorithm is fully unsupervised, and it proceeds as follows [2]. First, the text is tokenized and annotated with part of speech tags – a preprocessing step required to enable the application of syntactic filters. Here author consider only single words as candidates for addition to the graph, with multi-word keywords being eventually reconstructed in the post-processing phase.

Next, all lexical units that pass the syntactic filter are added to the graph, and an edge is added between those lexical units that co-occur within a window of words. After the graph is constructed (undirected unweighted graph), the score associated with each vertex is set to an initial value of 1.

5.3.TextRank for Sentence Extraction

To apply TextRank, we first need to build a graph associated with the text, where the graph vertices are representative for the units to be ranked. For the task of sentence extraction, the goal is to rank entire sentences, and therefore a vertex is added to the graph for each sentence in the text.

The co-occurrence relation used for keyword extraction cannot be applied here for sentence extraction , since the text units in consideration are significantly larger than one or few words, and "co-occurrence" is not a meaningful relation for such large contexts. Instead, author defined a different relation, which determines a connection between two sentences if there is a "similarity" relation between them, where "similarity" is measured as a function of their content overlap.The overlap of two sentences can be determined simply as the number of common tokens between the lexicalrepresentations of the two sentences, or it can be run through syntactic filters, which only count words of a certain syntactic category, e.g. all open class words, nouns and verbs, etc.Other sentence similarity measures, such as string kernels, cosine similarity, longest common subsequence, etc. are also possible.

5.4. Advantages and Limitation

In this paper , author clearly showed that the accuracy achieved by TextRank in these applications is more than that of previously proposed state-of-the-art algorithms. TextRank succeeds in identifying the most important sentences in a text based on information exclusively drawn from the text itself.

TextRank is fully unsupervised, and relies only on the given text to derive an extractive summary, which represents a summarization model closer to what humans are doing when producing an abstract for a given document.

An important aspect of TextRank is that it does not require deep linguistic knowledge, nor domain or language specific annotated corpora. So, due to this it is highly portable to other domains, genres, or languages.

Finally, another advantage of TextRank over previously proposed methods for building extractive summaries is that it does not require training corpora, which makes it easily adaptable to other languages or domains.

6. Automatic Summarization of Twitter Topics

6.1. Phrase Reinforcement Algorithm

PR algorithm[14] begins with a starting phrase, which is the topic for which one desires to generate a summary. These are typically a trending topic, but can be other non-trending topics as well. Given the returned set of posts, the algorithm next filters the posts to remove any spam or irrelevant posts. Filtering is an important step since spam and other irrelevant posts can mislead the PR algorithm into summarizing the spam instead of the desired content. The spam is filtered using a Naive Bayes classifier which they have trained using previously gathered spam content from ***Twitter.com*** and also remove any non-English posts as well as duplicate posts since they are going to create only English summaries .

The central idea of the PR algorithm is to build an ordered acyclic graph of all the words within the set of training posts to the algorithm. The graph is organized around a central root node, which contains the starting phrase of the summary. Adjacent to the starting node are words that occur either immediately before or after the starting

phrase within each of the training posts. These adjacent words are also placed either before or after the starting node respective of the ordering found within the training posts.

Once the graph is built, the PR algorithm begins searching for the best partial summary by summing the weight of every unique path starting from the root node to each of the leaf nodes. The path with highest weight is considered the best partial summary path from the root node.

6.2.Advantages and Limitations

PR algorithm may perform best when a topic has a dominant phrase pattern around the central topic. Whenever a topic is naturally part of a larger phrase, the PR algorithm works well and is able to isolate these dominant phrases from the set of input posts. This is especially true for #hashtag topics which is a convention twitter users have adopted in order to make certain topics easy to find via searching for the hashtag. If the hashtag does not naturally fall within a phrase, then the PR algorithm is not able to generate a dominant phrase around the topic.

7. Sequential Summarization: A New Application for Timely Updated Twitter Trending Topics

In this paper [12] ,author proposed Sequential summarization approch to generate the summary of trending topic on twitter. Here aims to generate a series of chronologically ordered subsummaries for a given Twitter trending topic. Each sub-summary is supposed to represent one main subtopic or one main aspect of the topic, while a sequential summary, made up by the subsummaries, should retain the order the information is delivered to the public. In such a way, the sequential summary is able to provide a general picture of the entire topic development.

7.1. Subtopic Segmentation

Here , author described One of the keys to sequential summarization is subtopic segmentation. How many subtopics have attracted the public attention, what are they,and how are they developed? It is very important because it provide the valuable and organized materials for more fine-grained summarization approaches. They proposed the following two approaches to automatically detect and chronologically order the subtopics.

7.1.1. Stream-based Subtopic Detection and Ordering

Typically when a subtopic is popular enough, it will create a certain level of surge in the tweet stream. The Offline peak area detection (Opad) algorithm is used to locate such surges by tracing tweet volume changes. It monitors the changes of the level of user attention.

7.1.2. Semantic-based Subtopic Detection and Ordering

One of the major drawback of stream-based approach is that, it fails to handle the cases where the posts about the same subtopic are received at different time ranges due to the difference of geographical and time zones. So to overcome this limitation they described semantic based approch.

The semantic-based subtopic detection approach breaks the time order of tweet stream, and regards each tweet as an individual short document. It takes advantage of Dynamic Topic Modeling to explore the tweet content.

7.2.Sequential Summary Generation

Once the subtopics are detected and ordered, the tweets belonging to each subtopic are ranked and the most significant one is extracted to generate the sub-summary regarding that subtopic. Two different ranking strategies are adopted to conform to two different subtopic detection mechanisms. For a tweet in a peak area, the linear combination of two measures is considered to evaluate its significance to be a sub-summary:

1.Subtopic representativeness measured by the cosine similarity between the tweet and the centroid of all the tweets in the same peak area;

2. Crowding endorsement measured by the times that the tweet is re-tweeted normalized by the total number of re-tweeting. With the DTM model, the significance of the tweets is evaluated directly by word distribution per subtopic. MMR is used to reduce redundancy in sub-summary generation.

7.3.Advantages and Limitations

The combination of the stream-based approach and the semantic-based approach leads to sequential summaries with high coverage, low redundancy, and good order.

8.Conclusion

Here, in this survey paper we have tried to describe various Single Document Summarization as well as Multi-document Summaraization Approch. We can conclude that each approch has its own significance and importance in genertaing the summary. Each summarization approch uses different algorithm and evaluation technique for measuring and comparing the accuracy of generated summary. So from all above described work we can conclude that, several challenges arise when we attempt to perform summary of tweets.

1) Words are often misspelled in tweets which means that we cannot use a dictionary or knowledge-base.

2) Many tweets are noisy and irrelevant to topic.

3) Twitter topic summarization must handle various numerious, short, dissimilar and noisy nature of tweet that causes poor performance .

9. References

[1] D. Radev, H. Jing, M. Styś, and D. Tam, “Centroid-based summarization of multiple documents,” Inf. Process. Manage., vol. 40, pp.919–938, 2004.

[2] R. Mihalcea and P. Tarau, “TextRank: Bringing order into texts,” in Proc. EMNLP-04, 2004, pp. 404–411.

[3] X. Wan and J. Yang, “Multi-document summarization using clusterbased link analysis,” in Proc. SIGIR-08, 2008, pp. 299–306.

[4] B. Sharifi, M.-A. Hutton, and J. Kalita, “Summarizing microblogs automatically,” in Proc. HLT/NAACL-10, 2010.

[5] B. Sharifi, M.-A. Hutton, and J. Kalita, “Experiments in microblogsummarization,” in Proc. IEEE 2nd Int. Conf. Social Comput., 2010.

[6] D. Inouye, Multiple post microblog summarization REU Research Final Rep., 2010.

[7] G. Erkan and D. Radev, "LexRank: Graph-based Centrality as Salience in Text Summarization," J. Artif. Intell. Res., vol. 22, pp. 457–479,2004

[8] W. Cohen, V. Carvalho, and T. Mitchell, "Learning to classify email into 'speech acts'," in Proc. EMNLP-04, 2004, pp. 309–316.

[9] R. Zhang, D. Gao, and W. Li, "What are tweeters doing: Recognizing speech acts in twitter," in Proc. AAAI-11 Workshop Analyzing Microtext, 2011.

[10] A. Ritter, C. Cherry, and B. Dolan, "Unsupervised modeling of twitter conversations," in Proc. HLT-NAACL-10, 2010, pp. 172–180.

[11] M. Jeong, C.-Y. Lin, and G. Lee, "Semi-supervised speech act recognition in emails and forums," in Proc. EMNLP-09, 2009, pp. 1250–1259.

[12] Dehong Gao, Wenjie Li and Renxian Zhang."Sequential Summarization: A New Application for Timely Updated Twitter Trending Topics.",51 ACL Sofia, Bulgaria, August 4-9, 2013

[13] B. Han and T. Baldwin, "Lexical normalisation of short text messages:Makn sens a #twitter," in Proc. ACL-11, 2011, pp. 368–378.

[14] B. Sharifi, M.-A. Hutton, and J. Kalita, "Automatic Summarization of Twitter Topics," in National Workshop on Design and Analysis of Algorithm, Tezpur, India, 2010.

[15] X. Yang, A. Ghoting, Y. Ruan, S. Parthasarathy. "A Framework for Summarizing and Analyzing Twitter Feeds." ACM SIGKDD IntÂl Conference on Knowledge Discovery and Data Mining, 2012.

[16] U. Hahn and I. Mani, "The challenges of automatic summarization,"Computer, pp. 29–36, 2000.

[17] Chao Shen, Fei Liu, Fuliang Weng, and Tao Li , "A Participant-based Approach for Event Summarization Using *Twitter Streams,"In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (NAACL 2013)