Topic based hierarchical summarisation of Twitter

Bushra Siddique* and Nadeem Akhtar

Department of Computer Engineering,
Aligarh Muslim University,
Aligarh, India
Email: bushrasiddique006@gmail.com
Email: nadeemalakhtar@gmail.com
*Corresponding author

Abstract: Twitter has become a rich source of information nowadays. The data generated however is so large in volume that it is not possible to manually go through each and every tweet to understand the context of data. One of the ways to get insight into the bulk of data at hand is to know the topics contained in it. As in the context of Twitter, we define topics to be long-lasting subjects around which the conversations of people revolve, such as sports, music and politics amongst others. However, the topics identified may be large in number and might be cumbersome for human interpretation. Considering these views, in this paper we address the information overload problem of Twitter data and propose a topic based hierarchical summarisation framework for the same. In contrast to imposing restrictions on topic models to depict the hierarchical structure, we propose an algorithm which constructs a topic hierarchy out of any given number of topics. We showcase the effectiveness of the proposed algorithm for the Twitter dataset prepared for Egyptair MS181 flight incident.

Keywords: Twitter; topic based summarisation; topic hierarchy; hierarchical summarisation.

Reference to this paper should be made as follows: Siddique, B. and Akhtar, N. (2019) 'Topic based hierarchical summarisation of Twitter', *Int. J. Spatio-Temporal Data Science*, Vol. 1, No. 1, pp.70–83.

Biographical notes: Bushra Siddique received her BTech in Computer Engineering and MTech in Software Engineering from Aligarh Muslim University, India. She is currently working as a Research Scholar in the Department of Computer Engineering, AMU. Her research interest includes data mining, information retrieval and soft computing.

Nadeem Akhtar received his BTech and MTech in Computer Engineering at Aligarh Muslim University, Aligarh, India. He has been working as an Assistant Professor in the Department of Computer Engineering, AMU for the last 13 years. His research interests include soft computing, data analytics and text mining.

This paper is a revised and expanded version of a paper entitled 'Topic Based Hierarchical Summarization of Twitter' presented at International Conference on Smart Technologies in Computer and Communication (SmartTech-2017), Amity Institute of Information Technology, Amity University, Rajasthan, India, 27–29 March 2017.

1 Introduction

Although rich in content, the data generated from Twitter is very high in volume. It is reported that every second, on average, Twitter sees around 6,000 tweets, which corresponds to over 350,000 tweets sent per minute, 500 million tweets per day and around 200 billion tweets per year. Owing to such a high volume, it would require a substantial amount of effort to extract the meaningful information manually which is not feasible. This fuels the need for a summarisation framework which could help in exploring the voluminous data with minimal effort.

As for Twitter data, its major characteristics include the topics and the events. In the literature, events are identified as real world occurrences that unfold over time and space and attract short term attention of the crowd whereas topics are identified as long-lasting subjects around which the conversations of people revolve. Works Akhtar and Siddique (2017, 2018) dedicated to hierarchical summarisation of events in Twitter have been done. In this paper, however, we focus only on topics and aim to summarise the Twitter data hierarchically leveraging the benefits of the hierarchical summarisation.

The problem addressed in the paper is as follows: Given a corpus of Twitter data, a framework for topic based hierarchical summarisation is implemented. The resulting topic hierarchy is a multilevel tree structure where each node corresponds to a topic. The nodes at higher levels correspond to general concepts whereas the nodes at lower levels correspond to specific concepts. The parent child relationship between nodes is such that the topic of the child node represents a sub topic of the parent node topic. The Twitter data thus summarised hierarchically through the topic hierarchy offers dual advantages. Firstly, it does not overwhelm the user and discloses the information progressively. Secondly, the user has the freedom to navigate or to traverse to only those segments of the hierarchy which are of interest to him.

In general, topic based text summarisation has proven to be an effective technique. However, if the number of topics identified is large in number, it is cumbersome and challenging for human interpretation. To cope up with this, a number of studies have come up with the fact that, the identified topics when arranged in a hierarchical fashion present a more comprehendible solution. Hierarchical topic models are one way of identifying topic hierarchies for given data. For instance the hLDA model proposed by Blei et al. (2004), successfully accommodates growing data and identifies topic hierarchies. However, the depth of the hierarchy needs to be predefined. Also, the top level topics identified by hLDA usually consist of stopwords which might be less human interpretable. In this work, we do not impose restrictions on topic models to identify the hierarchical structure of topics. Rather, we make use of a topic model to identify topics and prepare a hierarchical structure for the identified topics through our proposed algorithm.

The rest of the paper is organised as follows. In Section 2, we discuss the related work, in Section 3 we present the detailed framework, in Section 4 we discuss our experimental study and finally the paper is concluded.

2 Related work

2.1 Hierarchy construction for Twitter data

Studies in Akhtar and Siddique (2017, 2018) are targeted on hierarchical visualisation of event detection in Twitter. In Akhtar and Siddique (2018), authors have proposed a novel unified workflow in which events are detected and a hierarchy of the detected events is generated through recursive hierarchical clustering. The levels of hierarchy represent the timeline at different granularities of time. The same idea is extended to carry out a case study for a sport event in Akhtar and Siddique (2017).

In Zhu et al. (2015), authors have proposed a strategy for constructing timeline coherent topic hierarchies for performing evolutionary analysis on microblog messages. For the purpose, they have considered batch of messages together and used biterm topic model for extracting coherent topics from each batch. Further, through Bayesian Rose Trees they have prepared a hierarchical structure out of the extracted topics and devised a cross- tree random walk for linking each pair of trees into a timeline hierarchy.

Gu et al. (2011) present an approach known as ETree (Event Tree) for event modelling. The idea they have employed is based on use of an n-gram based technique for grouping messages related to a particular event into semantically-coherent information blocks. Further, they have constructed hierarchical structure of themes through an incremental modelling process, and utilised a life cycle-based temporal analysis methodology for identifying possible causal relationships amongst information blocks.

2.2 Topic hierarchy construction

Studies like in Dou et al. (2012, 2013) have come up with visualisation systems for data exploration. In Dou et al. (2012), authors have proposed an interactive visualisation system called Leadline for automatic event detection and exploration for news and social media data. The main focus of the system is to investigate 4Ws, namely, who, what, when, and where for each event through topic modelling (LDA), event detection, and named entity recognition techniques. For the purpose of visualisation, the authors have made use of Hellinger distance to place similar topics close to each other. In Dou et al. (2013), authors have presented a visual analytics system called HierarchicalTopics for exploring large text corpora. The proposed system generates initial topic hierarchy through a computational algorithm topic rose tree, based on concepts of Bayesian Rose Tree. Further, the user has the freedom to change the hierarchical structure based on his mental model through visual interaction.

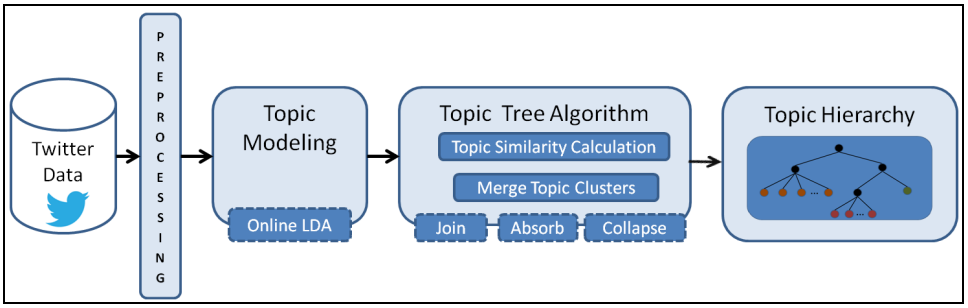
In Marcacini and Rezende (2010), authors have proposed an algorithm incremental hierarchical term clustering (IHTC) for incremental construction of topic hierarchies. IHTC uses co-occurrence between terms to perform document clustering and generates a dendrogram depicting the clustering.

In Wang et al. (2013), a phrase-centric framework known as CATHY (Constructing A Topical HierarchY) is proposed for generating topic hierarchy through recursive clustering and ranking targeted for short, content-representative texts. The topics over multiple levels are represented by ranked lists of phrases.

3 The framework

The block diagram for the proposed framework is shown in Figure 1. From the figure three key stages could be identified. At first stage, the collected data undergoes preprocessing followed by the next stage of topic modelling and finally the last stage of hierarchy construction.

Figure 1 The framework (see online version for colours)



3.1 Preprocessing

Tweeting is but a social activity. As a result, the data generated can be unpredictably noisy. We undertake following steps for preprocessing tweets:

- escaping HTML characters
- tokenisation using Twokeniser tool (Krieger and Ahn, 2010)
- removal of stop-words
- removal of expressions
- removal of URL
- removal of mentions
- identifying hashtags
- slangs lookup.

After the preprocessing is done, we discard tweets that have less than 2 terms.

3.2 Topic modelling

The topic modelling stage identifies topics from the given cleansed data. From literature, there exists a number of methods to identify topics from a given data collection. It is worth mentioning that the choice of method in the pipeline shown is rather flexible. For implementation purpose, we have make use of Online LDA (Bach and Blei, 2010) for extracting topics.

3.3 Topic tree algorithm

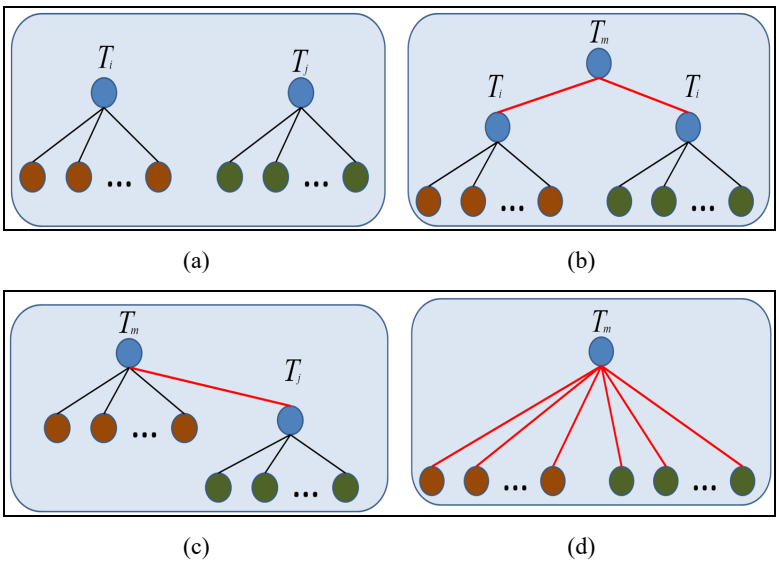
It is known that the binary tree assumes a maximum number of two branches at every node. Since any number of topics can be grouped together, binary tree may not represent the true picture of the relation between topics in topic hierarchy. To cope up with this limitation we make use of multi-way tree or rose tree. The key idea of the algorithm is based on the concept of Bayesian rose trees (BRT) (Blundell et al., 2012). BRT generates a hierarchy through hierarchical clustering using three operations namely join, absorb and collapse. Since the nodes in our case are topic vectors, we adapt the idea of Bayesian rose trees making it applicable to the results of topic modelling.

3.3.1 Non-incremental topic tree algorithm

Given two individual trees T_i and T_j as shown in Figure 2(a), they can be merged through either one of the three operations namely join, absorb, collapse as shown in Figures 2(b), 2(c) and 2(d) respectively. The operations are defined below.

- *Join*: We choose join operation for merging when the branches of T_i and T_j are related to each other but are distinguishable enough to exist separately. Mathematically, given T_i and T_j, $T_m = \{T_i; T_j\}$, i.e., T_m has two branches.
- *Absorb*: We choose absorb operation for merging when the branches of T_i and T_j are related to each other but T_j denotes some finer distinguishing feature. The root of the merged tree is denoted as T_m. Mathematically, given T_i and T_j, $T_m = \{branch(T_i); T_j\}$, i.e., T_m has $|num\_branch(T_i) + 1|$ branches.
- *Collapse*: We choose collapse operation for merging when the branches of T_i and T_j are indistinguishable and hence must be combined. The root of the merged tree is denoted as T_m. Mathematically, given T_i and T_j, $T_m = \{branch(T_i), branch(T_j)\}$, i.e., T_m has $|num\_branch(T_i) + num\_branch(T_j)|$ branches.

Figure 2 Operations of non incremental topic tree algorithm (see online version for colours)



(a) (b)

(c) (d)

For measuring similarity between two given topics T_a and T_b, we choose Hellinger distance as the similarity metric. Intuitively, lower is the value of the Hellinger distance; higher is the similarity between two topics. Furthermore, the probability distribution of a node containing child nodes is computed as the average of the distribution of all child nodes.

Algorithm 1 Non-incremental topic tree

Input: $S = \{X_i, v\}$, $i = 1, 2, \ldots, n$; *v is the vocabulary of the corpus*
Output: Topic Tree, the topic hierarchy for the corpus
1 *Initialise* $T_i = \{X_i\}$, $i = 1, 2, \ldots, n$
2 *Initialise* $C = n$, *as count of clusters*
3 ***while*** $\boldsymbol{C} > 1$ ***do***
4 ***for*** *pair of trees* T_i *and* T_j *with minimum distance* ***do***
5 *Calculate cost*(T_i, T_j) *for the operations join, absorb and collapse using Algorithm 2*
6 *Find the operation op which yields lowest cost*
7 *Merge* T_i *and* T_j *into* T_m *using operation op*
8 *Replace* T_i *and* T_j *with the new merged tree* T_m
9 *Update* $C = C - 1$
10 *end* ***for***
11 *end* ***while***

Algorithm 2 Cost of operations: join, absorb and collapse

Input: Tree T_i *and* T_j
Output: $cost(T_i, T_j)_{join}$, $cost(T_i, T_j)_{absorb}$ *and* $cost(T_i, T_j)_{collapse}$
1 $T_{(m_{join})} = join(T_i, T_j)$
2 $T_{(m_{absorb})} = absorb(T_i, T_j)$
3 $T_{(m_{collapse})} = collapse(T_i, T_j)$
4 $cost(T_i, T_j)_{join} = D(T_{(m_{join})}, T_i) + D(T_{(m_{join})}, T_j)$
5 $cost(T_i, T_j)_{absorb} = D(T_{(m_{absorb})}, T_i) + D(T_{(m_{absorb})}, T_j)$
6 $cost(T_i, T_j)_{collapse} = D(T_{(m_{collapse})}, T_i) + D(T_{(m_{collapse})}, T_j)$

Algorithm 1 takes as input the set S of n topics obtained from the topic modeling stage and generates the topic tree depicting the topic hierarchy as the output. Each topic is denoted by the probability distribution X_i of the terms over the vocabulary. In lines 1–2, each topic is initialised to a tree containing a single node forming n clusters. In lines 3–11, pair of trees is iteratively merged till only a single tree is left. Line 4 makes use of similarity metric to find the pair of trees which are most similar. Line 5 calculates the cost of merging the pair of trees for all the three operations (join, absorb, collapse) as per Algorithm 2. In lines 6–9, the operation with lowest cost is selected, merging is

performed and accordingly the cluster count is reduced by 1. The operation with the lowest cost is selected to ensure that the tree obtained after merging has minimal variation with respect to the given pair of trees. Next, we present the algorithm for calculating the cost of merging two trees for the three operations: join, absorb, collapse.

Algorithm 2 takes as input the pair of trees to be merged and returns the cost of merging for all the three possible operations. Lines 1, 2 and 3 find the node after merging the pair of trees via join, absorb and collapse respectively. Lines 4-6 calculate the cost of respective operation making use of the similarity metric. Intuitively, the merging operation that produces the node most similar to the given pair of trees is the lowest cost operation.

The complexity of the topic tree algorithm is the same as the BRT algorithm. At the first step, the distance between every pair of topics needs to be computed. For n topics, there are $O(n^2)$ such pairs. After that, these pairs must be sorted for finding the smallest distance. This further requires $O(n^2 logn)$ computational complexity.

3.3.2 Incremental topic tree algorithm

Given an existing tree T and a new topic T_n as shown in Figure 3(a), we formulate two operations to add the new topic node to the existing hierarchy. The two operations are iJoin and iAbsorb, and are based upon the join and absorb operation defined for Non Incremental Topic Tree algorithm. The idea is that we find the closest subtree in the existing tree and add the new topic node to it. The algorithm for finding closest subtree is outlined in Algorithm 3. Let the root of the closest subtree be Ts as shown in Figure 3(b). We now define the operations iJoin and iAbsorb.

- *iJoin*: We initialise the new topic node as an individual tree and merge the closest subtree and the new topic tree via join operation. Since the subtree is part of an existing tree, we update the branches of parent of T_s accordingly. The iJoin operation is depicted in Figure 3(c). The parent node of T_s in the existing tree is denoted as T_p and the new node after join operation is denoted as T_m. The branch T_s of T_p is replaced by T_m accordingly. The updating of branches needs to be done up to the root of the existing tree.
- *iAbsorb*: We merge the closest subtree and the new topic node via absorb operation. The iAbsorb operation is depicted in Figure 3(d). The new topic node is simply added as one of the branches of T_s. The updating of branches needs to be done up to the root of the existing tree.

Algorithm 3 Incremental topic tree

Input: T, the existing Topic Tree and T_n, the topic node to be added

Output: Updated Topic Tree, the topic hierarchy for the corpus

1 *Initialise T_n as the individual tree containing a single node*

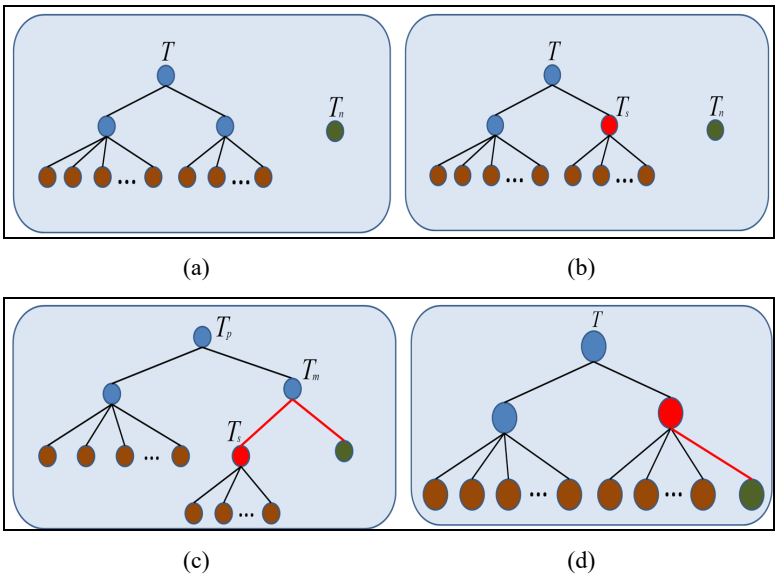
2 *T_s = find closest subtree(T, T_n) using Algorithm 4*

3 *Calculate cost(T_n, T_s) for the operations iJoin and iAbsorb using Algorithm 5*

4 *Find the operation op which yields lowest cost*

5 *Add T_n to T_s using operation op*

Figure 3 Operations of incremental topic tree algorithm (see online version for colours)



Algorithm 3 takes as input the existing topic hierarchy and the new topic node to be added and generates the updated topic hierarchy. Line 1 initialises the new topic node as an individual tree containing single node. In line 2, the subtree closest to the node to be added is found as per Algorithm 4. In lines 3–5, the cost of the two possible operations is calculated via Algorithm 5 and the one with lower cost is performed. The algorithm for finding closest subtree is given below.

Algorithm 4 Finding closest subtree T_s

Input: Existing Tree T, Tree node T_n to be added
Output: Subtree T_s closest to T_n
1. for all leaf nodes T_l of T
2. Find $T_{(l,min)}$ such that D(T_l, T_n) is minimum
3. for all nodes T on the path from $T_{(l,min)}$ to root(T)
4. Find T_{min} such that D(T, T_n) is minimum
5. Return T_{min}

Algorithm 4 takes as input the existing tree T and the new topic node T_n to be added, and finds the subtree T_s closest to T_n. The leaf nodes of the tree are pure topics obtained from the topic modelling stage. Lines 1–2 find the leaf node, i.e., the topic that is closest to the new topic node. After the closest leaf is obtained, lines 3–4 traverse the path from that leaf node to the root to find the node which is closest to the new topic node. The algorithm for calculating the cost of operations: iJoin and iAbsorb is shown in Algorithm 5.

Algorithm 5 Cost of operations: iJoin and iAbsorb

Input: Node T_n and Subtree T_s

Output: cost$(T_n, T_s)_{iJoin}$ and cost$(T_n, T_s)_{iAbsorb}$

1 $T_{(m_{iJoin})} = iJoin(T_n, T_s)$

2 $T_{(m_{iAbsorb})} = iAbsorb(T_n, T_s)$

3 $cost(T_n, T_s)_{iJoin} = D(T_{(m_{iJoin})}, T_n) + D(T_{(m_{iJoin})}, T_s)$

4 $cost(T_n, T_s)_{iAbsorb} = D(T_{(m_{iAbsorb})}, T_n) + D(T_{(m_{iAbsorb})}, T_s)$

Algorithm 5 takes as input the new topic node T_n and the subtree T_s to which the addition is to be made, and calculates the cost of adding for the two possible operations. Lines 1 and 2 find the node after adding the new node to the subtree via iJoin and iAbsorb respectively. Lines 3–4 calculate the cost of respective operation making use of the similarity metric.

3.4 Topic summarisation

Corresponding to each node in the topic hierarchy, following set of information is provided for the purpose of topic summarisation.

- *Keywords*: Every node in the topic hierarchy corresponds to a topic. Further, every topic is a probability distribution over words. We list top *m* terms based on the values of term probability for each node.
- *Representative tweets*: Since, tweets are short in length; the content of one tweet might not provide complete information. Hence, we extract at least *n* tweets, ($n > 1$), for each topic node. The extraction method is based on matching the keywords of the topic node with the tweet content and selecting those tweets which contain the maximum number of keywords.

4 Experimental study

4.1 Dataset preparation

We extracted Twitter data making use of the public streaming API of Twitter for the major event that occurred on 29th March, 2016: Egyptair MS181 flight incident. The dataset contained 514,902 English language tweets.

4.2 Analysis of dataset

Since there is no way of identifying the exact number and nature of topics in the dataset, we relied on the mainstream news articles to observe that the dataset actually contained 9 meaningful topics. The observed topics are given below:

A arrest

B release

C wife

D letter

E hijack

F ministry opinion

G presidential statement

H selfie

I explosives.

With the help of an expert, we performed the logical groupings of the identified topics and constructed a topic hierarchy which is as shown in Figure 4. We label this hierarchy as 'hierarchy 1' for further reference. The labels in the nodes of the tree correspond to the topics in the list mentioned above. For instance, as shown in the Figure 4, the nodes E and I are grouped together. Correspondingly, the topics 'hijack' and 'explosives' are grouped together.

Figure 4 Hierarchy 1: ground truth topic hierarchy

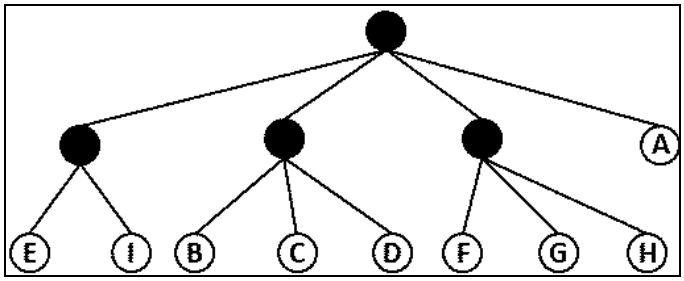
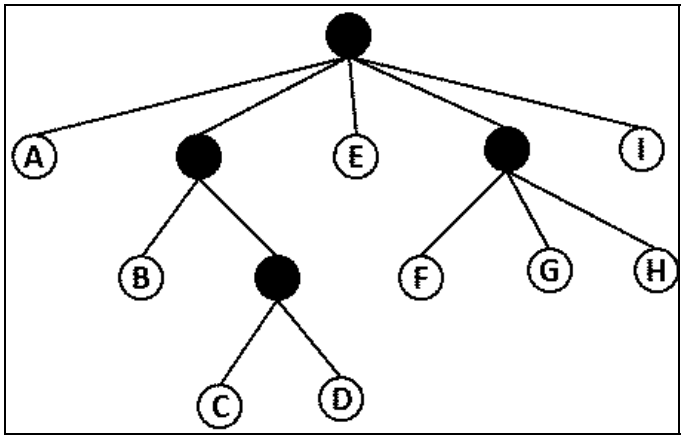


Figure 5 Hierarchy 2: non incremental topic hierarchy

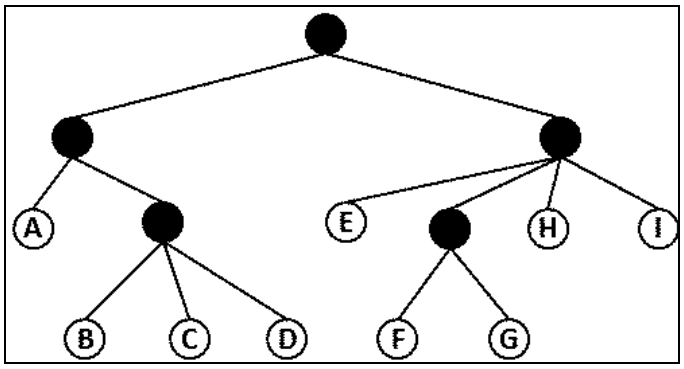


4.3 Results

Two versions of the topic tree algorithm for topic hierarchy construction are presented, the first one being non incremental and the other one being incremental. The non-incremental algorithm takes all the nine topics and constructs a hierarchy out of them. The resultant hierarchy is shown in Figure 5. We label this hierarchy as 'hierarchy 2' for further reference. Furthermore, the incremental version takes five topics to construct the initial hierarchy and add the rest of the four topics incrementally.

The resultant hierarchy is shown in Figure 6. Again, we label this hierarchy as 'hierarchy 3' for further reference.

Figure 6 Hierarchy 3: incremental topic hierarchy



4.4 Evaluation measure

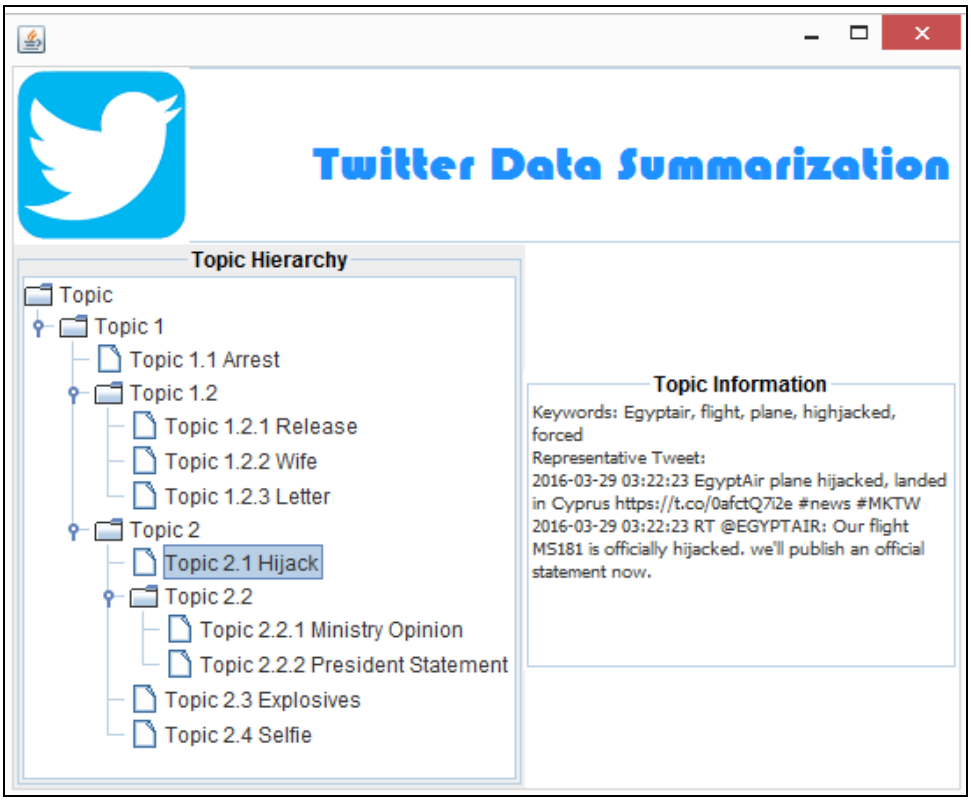
The idea is to estimate the extent to which the logical groupings performed as a result of the Topic Tree Algorithm is similar to the one performed manually. In order to find similarity, we examine all sibling pair of the direct topics in the hierarchy. Since the topic pair defines the logical groupings, this comparison will provide a way to measure how similar one hierarchy is to the other.

In the simplest sense, we can count the number of siblings pair in the two hierarchies that are overlapping. Since, the two hierarchies may contain different number of siblings pair, this number may not provide the actual similarity measure. As a result, a ratio comparing the number of overlapping pairs to the total number of pairs in one of the hierarchies is used.

Keeping the above discussion in mind, we now define the similarity metric α. Given topic hierarchy A and B, and calculating the similarity of A to B, α is calculated as under:

$$\alpha = \frac{Count\ of\ overlapping\ sibling\ pairs\ in\ A\ and\ B}{Total\ Count\ of\ sibling\ pair\ in\ B}$$

Figure 7 Topic summary for the topic 'hijack' in incremental topic hierarchy (hierarchy 3) (see online version for colours)

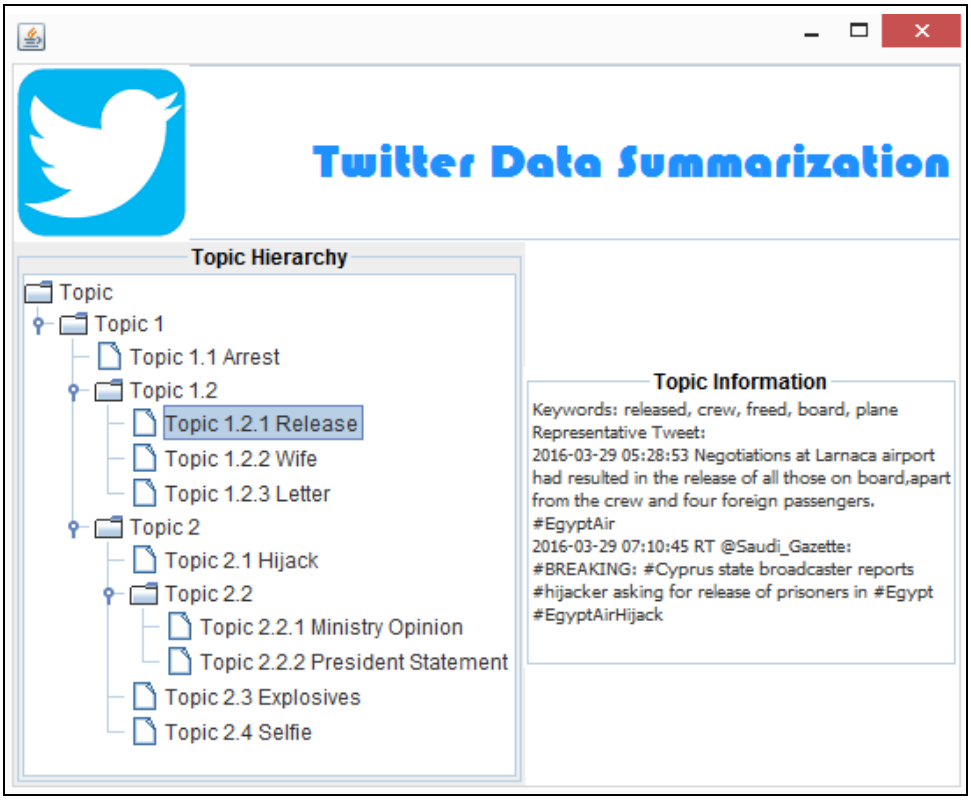


4.5 Results and discussions

The total number of siblings pair in Hierarchy 1 is 7, some of which include (*Hijack*, *Explosives*), (*Release*, *Letter*), (*Ministry Opinion*, *Selfie*) and so on. Coincidentally, the total number of sibling pair in Hierarchy 1 and 2 is also 7. The number of overlapping pairs for Hierarchy 1 and 2 as well as Hierarchy 1 and 3 is 5. This amounts to α having the value of ~70%. The value of the similarity metric so obtained is favourable. We thus conclude that the proposed algorithm performs logical groupings of the identified topics similar to the one manually performed.

Also, we discussed earlier that as a part of the framework, a topic summary for each topic is generated which consist of a set of keywords and a couple of representative tweets. The snapshots in Figures 7 and 8 shows the topic summary generated corresponding to topic 'hijack and release' for hierarchy 3 respectively.

Figure 8 Topic summary for the topic 'release' in incremental topic hierarchy (hierarchy 3) (see online version for colours)



5 Conclusions

It is known that the data generated from Twitter contains timely information of all kinds. Keeping in view its bulk volume, we propose a topic based hierarchical summarisation framework for Twitter data. As a part of framework, we generate a hierarchy based on topics compounded with a set of keywords and few representative tweets describing the topics. Furthermore, the hierarchy of topics is not generated through predefined hierarchical topic models, rather a novel proposed algorithm which prepares the hierarchical structure out of the topics identified through any topic model. We present a case study on the prepared dataset and demonstrate the efficacy of the framework.

References

Akhtar, N. and Siddique, B. (2017) 'Hierarchical visualization of sport events using twitter', *Journal of Intelligent and Fuzzy Systems*, Vol. 32, No. 4, pp.2953–2961.

Akhtar, N. and Siddique, B. (2018) 'On hierarchical visualization of event detection in twitter', in *Advances in Computer and Computational Sciences*, pp.571–579, Springer.

Bach, F. and Blei, D.M. (2010) *Online Learning for Latent Dirichlet Allocation*, NIPS, Hofmann.

Blei, D.M., Jordan, M.I., Griffiths, T.L. and Tenenbaum, J.B. (2004) 'Hierarchical topic models and the nested Chinese restaurant process', in *Advances in Neural Information Processing Systems*, pp.17–24.

Blundell, C., Teh, Y.W. and Heller, K.A. (2012) *Bayesian Rose Trees*, arXiv preprint arXiv:1203.3468.

Dou, W., Wang, X., Skau, D., Ribarsky, W. and Zhou, M.X. (2012) 'Leadline: interactive visual analysis of text data through event identification and exploration', in *2012 IEEE Conference on Visual Analytics Science and Technology (VAST),* pp.93–102, IEEE.

Dou, W., Yu, L., Wang, X., Ma, Z. and Ribarsky, W. (2013) 'Hierarchicaltopics: visually exploring large text collections using topic hierarchies', *IEEE Transactions on Visualization and Computer Graphics*, Vol. 19, No. 12, pp.2002–2011.

Gu, H., Xie, X., Lv, Q., Ruan, Y. and Shang, L. (2011) 'Etree: Effective and efficient event modeling for real-time online social media networks', in *2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, Vol. 1, pp.300–307, IEEE.

Krieger, M. and Ahn, D. (2010) 'Tweetmotif: exploratory search and topic summarization for twitter', in *Proc. of AAAI Conference on Weblogs and Social*, Citeseer.

Marcacini, R.M. and Rezende, S.O. (2010) 'Incremental construction of topic hierarchies using hierarchical term clustering', in *SEKE*, p.553.

Wang, C., Danilevsky, M., Desai, N., Zhang, Y., Nguyen, P., Taula, T. and Han, J. (2013) 'A phrase mining framework for recursive construction of a topical hierarchy', in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.437–445, ACM.

Zhu, J., Li, X., Peng, M., Huang, J., Qian, T., Huang, J., Liu, J., Hong, R. and Liu, P. (2015) 'Coherent topic hierarchy: a strategy for topic evolutionary analysis on microblog feeds', in *International Conference on Web-Age Information Management*, pp.70–82, Springer.