Use of the Pole-based Overlapping Clustering Method for Labeling Tweets @ IRMiDis FIRE 2017

Noushin Fadaei
Hildesheim University
Hildesheim, Germany
fadaei@uni-hildesheim.de

Thomas Mandl
Hildesheim University
Hildesheim, Germany
mandl@uni-hildesheim.de

1 INTRODUCTION

Real-time information in social media such as *Twitter* on major events, for example major earthquakes or terrorist attacks, plays an important role in revealing critical needs resulted by the disaster and the resources to address them; however this information is also influentially clouded by emotional posts. Finding the critical information and finding their relevance to the resources would provide a valuable platform for further analysis.

This working note covers the method we used to tackle the challenge exposed to discussion by FIRE 2017 track: Information Retrieval from Microblogs during Disasters (IRMiDis) [1]. The first task is to identify tweets regarding the *required* resources such as food, water or medical aids and the *available* or *potential available* resources like the already transported or distributed resources among a dataset of over 50,000 tweets regarding the Nepal-India earthquake in 2015.

A tweet can represent the needs for resources as well as the availability of some others at specific locations; therefore it would be reasonable to use a soft clustering algorithm which shows the relevance degree of one tweet to each of the classes. Pole-based overlapping clustering algorithm utilizes this notion to not only detect the proper class for each object but also let the objects share various classes in case they are similar enough to each class.

2 METHODOLOGY

2.1 Pole-based Overlapping Clustering (PoBOC)

The algorithm starts off with a similarity matrix $X \times X$ containing the similarity of each object with all the other objects. Based on this matrix a similarity graph is built, where an edge exists between two objects, if the similarity of these two is greater than the average similarities of each of the two individual objects and the whole set of objects. [2]
In the next step the poles representing the individual medoids of the clusters are built. For this purpose the cliques within the similarity graph are obtained. These cliques form subgraphs where each node/vertex is linked to all other nodes/vertices, starting off with a node with at least the degree of one, with the lowest average similarity to all other objects. The starting nodes for the next poles are determined in order to minimize the similarity with previous poles. The termination criterion for this process is again based on the average similarity: if the average similarity of the starting node for the new pole with the already built poles is smaller than the average similarity of that node with all other objects, then the process is ended and the number of clusters based on the previously found poles is determined. [2]
Consequently all objects which are not part of the poles are assigned to the clusters based on their similarity with the poles. Based on those similarities a ranking of the poles is generated for each object and it is always assigned to the most similar one (rank 1). In order to validate the assignment of the clusters on the other ranking positions the following condition is required: The average similarity between the object and the previous as well as the following pole needs to be bigger or equal to the similarity of the object to the current pole.
Following the hierarchical agglomerative clustering method "single-link" [4], the hierarchical clusters are being built until all objects fill in one single cluster. The algorithm is shown in figure 1.

2.2 Expert Knowledge as Relevance Feedback

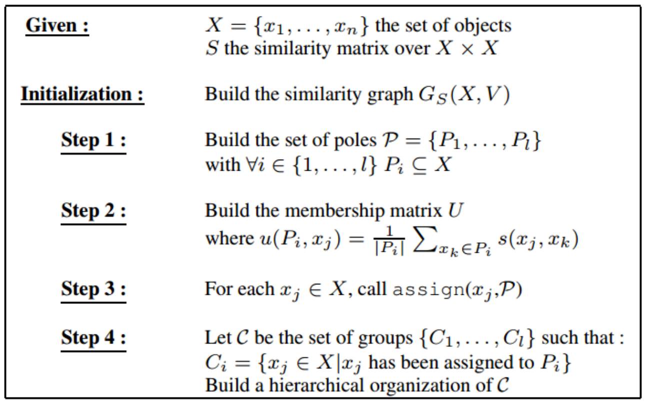
PoBOC builds a graph on top of the similarity matrices of the given objects and tries to find out the strongly connected graphs namely *poles*; however if we consider these poles are already identified by the experts (similar to relevance feedback methods), our method tries to find the objects that are close enough to each pole. One alternative solution is to cluster the entire data and consider the level of hierarchy that covers three cluster (as need class, availability class and class of irrelevant tweets). Although PoBOC supports overlapping clusters, it doesn't explain how to organize a soft clustering over two poles. An object belongs to the closest pole. In case an object is similar to both of the classes evenly, we considered it a member of both. The similarity of each object and a pole is calculated as follows:

$$s(x_i, P_m) = \frac{1}{|P_m|} \sum_{x_j \in P_m} s(x_i, x_j) \tag{1}$$

where x_i is an object and P_m is the m^{th} pole.

We have used *Weka* [3] for the pre-processing. We removed the links and used *Snowball* stop word list in addition to some few frequent words in the data collection namely earthquake, Nepal and death. As tweets are quite short the minimum term frequency of a word is not of a matter and for weighting tfidf is used to select the first 500 important words. These terms formed Weka instances which are taken as objects in PoBOC algorithm.

Developing this fuzzy algorithm, Euclidean distance is used as a similarity measure and the closer an object is to a pole, the higher it is ranked.



```
Given :            $X = \{x_1, \dots, x_n\}$ the set of objects
                   $S$ the similarity matrix over $X \times X$

Initialization :   Build the similarity graph $G_S(X, V)$

    Step 1 :       Build the set of poles $\mathcal{P} = \{P_1, \dots, P_l\}$
                   with $\forall i \in \{1, \dots, l\}\ P_i \subseteq X$

    Step 2 :       Build the membership matrix $U$
                   where $u(P_i, x_j) = \frac{1}{|P_i|} \sum_{x_k \in P_i} s(x_j, x_k)$

    Step 3 :       For each $x_j \in X$, call assign($x_j$,$\mathcal{P}$)

    Step 4 :       Let $\mathcal{C}$ be the set of groups $\{C_1, \dots, C_l\}$ such that :
                   $C_i = \{x_j \in X | x_j$ has been assigned to $P_i\}$
                   Build a hierarchical organization of $\mathcal{C}$
```

Figure 1: PoBOC: an overlapping clustering algorithm [2]

Table 1: Sub-task 1- Identifying need-tweets and availability-tweets

Submission Detail		Availability-Tweets Evaluation			Need-Tweets Evaluation			
Run ID	RunType	Precision@100	Recall@1000	MAP	Precision@100	Recall@1000	MAP	Average MAP
`Iwist_task1_1`	Automatic	0.0300	0.0194	0.0165	0.0400	0.1639	0.0417	0.0291

Source: Results of our methodology by IRMiDis - FIRE2017 [1]

3 RESULTS

The results of our method regarding task 1 are shown in Table 1. According to the evaluations of the organization committee, it ranked $13^t h$. The lack of lexical databases and semantic analysis in the pre-processing phase can explain the low performance of this method. Considering the amount of emotional stop words using sentiment analysis can also improve the representation of the vector objects.

In our method, we took the given training data as the experts feedback for which there are 674 tweets relevant to the availability class and less than one third of it namely 201 tweets relevant to the need class. Here the presumption is that the relevant tweets form a cluster meaning they have more similar terms overlap, however the relatively low recalls show the distribution of the relevant tweets does not match our assumption. Yet the recall of the need class is much better than the one of the availability class, and since we have less relevant tweets for the need class, it suggests the variety of the wordings regarding each class if not the variety of experts perspectives.

4 FUTURE WORK

Due to working on this task within a short time the semantic aspects of the training data and the co-appearance of the word were ignored. Also the relevance feedback methods can play an important role in case we look at the problem as an information retrieval task where training data can be used to produce relevant queries.

REFERENCES

[1] Moumita Basu, Saptarshi Ghosh, Kripabandhu Ghosh, and Monojit Choudhury. 2017. Overview of the FIRE 2017 track: Information Retrieval from Microblogs during Disasters (IRMiDis). In *Working notes of FIRE 2017 - Forum for Information Retrieval Evaluation (CEUR Workshop Proceedings).* CEUR-WS.org.

[2] Guillaume Cleuziou, Lionel Martin, and Christel Vrain. 2004. PoBOC: an Overlapping Clustering Algorithm. Application to Rule-Based Classification and Textual Data. In *In R. Lopez de Mantaras and L. Saitta, IOS Press, editor, Proceedings of the 16th European Conference on Artificial Intelligence.* 440–444.

[3] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* 11, 1 (Nov. 2009), 10–18. https://doi.org/10.1145/1656274.1656278

[4] A. K. Jain, M. N. Murty, and P. J. Flynn. 1999. Data Clustering: A Review. *ACM Comput. Surv.* 31, 3 (Sept. 1999), 264–323. https://doi.org/10.1145/331499.331504