Communication Overload Management Through Social Interactions Clustering

Juan Antonio Lossio-Ventura, Hakim Hacid, Mathieu Roche, Pascal Poncelet

Communication Overload Management Through Social Interactions Clustering

Juan Antonio Lossio-Ventura[1], Hakim Hacid[2], Mathieu Roche[3], Pascal Poncelet[1]
[1] LIRMM - University of Montpellier, France
[2] Zayed University, United Arab Emirates
[3] Cirad, TETIS & LIRMM, France

November 1, 2018

Abstract

We propose in this paper to handle the problem of overload in social interactions by grouping messages according to three important dimensions: (i) content (textual and hashtags), (ii) users, and (iii) time difference. We evaluated our approach on a Twitter data set and we compared it to other existing approaches and the results are promising and encouraging.

1 Introduction

It is very common in todays' social networks that several discussion threads around similar topics are opened at the same time in different distinct or overlapping communities. Being aware about these different threads may be difficult. Moreover, when new threads are created, it may be useful to provide the user with linked past tweets instead of generating new threads. Information linkage is the process by which different pieces of information are put together according to criteria and constraints to form a new information which is richer (*i.e.* increased) and which can be consumed by an user or automatically by another process.

This linkage can: (i) ease the digestion of information, *i.e.* its perception by users, (ii) enable a better information management from the system perspective, and (iii) allow other third-party applications to draw more benefits from a social content which, in a disparate form, is useless. The problem we are tackling can be formulated as follows: *Having a broad set of interactions between users of a social network with disparate messages and connections, how to link these interactions so that they are correlated consistently and significantly for either an end user or an automatic processor to navigate easier in this large content.*

To the best of our knowledge, our approach is the only one combining: (i) semantic, (ii) user and (iii) temporal dimensions to generate connections between short messages in social networks (*i.e.* in our work on Twitter), as we did in [11]. This process will

also allow to perform well other tasks, such as query recommendation [24], text understanding [10] (*i.e.* summarization), and event detection.

2 Related Work

Techniques proposed in this paper are mainly related to clustering of short text, and mire specially to clustering of tweets. One challenge in clustering short text is the sparse data problem, *i.e.* the exact keyword matching may not work well. So, traditional classification methods such as "Bag-Of-Words" have limitations. Thus, to solve this problem, there exist approaches mainly based on feature expansion: (i) expansion of text representation by exploiting related text documents, and (ii) expansion of features by bringing external information from knowledge bases. For (i), the objective is in extracting context information through search engines [1, 18, 5]. The enriched short texts may be seen as a long texts to be classified with approaches for long text clustering. This approach is not really appropriate for some online applications as it is highly time consuming and heavily depends on the search engines quality [20]. For (ii), the expansion is performed by augmenting with external information from knowledge bases such as Wikipedia, BabelNet, WordNet, DBPedia [13, 9, 21, 3, 19]. These techniques allow to obtain a set of explicit, or implicit, topics and then to connect the short text according to these topics. The use of known topics decreases the dependence on search engines. However, a possible issue is that the known topics may not be available for some applications [4].

The idea of linking social interactions has been discussed before. A related study, focused on electronic mails, detects conversations in email messages by grouping them in consistent collections [7]. Other studies focus on tweets clustering. For instance, Sriram et al. [17] address the problem of classification of tweets following a supervised machine learning approach. Messages are classified into five categories: *News (N), Events (E), Opinion (O) Deals (D) Private Messages (PM)* [15]. Most of the work in related to the problem of tweets clustering take into account the textual part, eventually enriched with an external knowledge. There is, to our knowledge, no methodology that takes into account the textual, hashtags, users and temporal aspect [14, 16, 23].

3 New Similarity Measure

In this section, we describe how to compute the new similarity measure for clustering of tweets.

3.1 Content Similarity (CS)

Social messages are by nature short, *e.g.* Twitter allows only 140 characters. As a consequent, it is usual hard to compute a textual similarity between such kind of messages because they might not have any keywords in common. In this case, traditional measures such as *Cosine, Overlap*, and *Jaccard* perform with poor results. In an attempt to overcome this problem, we propose to rely on a graph that captures the similarity be-

tween keywords instead of only co-occurrences. We propose a combination of a textual similarity (*txt*) and the hashtags (*hash*) that appear in tweets.

3.1.1 Textual Similarity (txt)

Before proceeding with the similarity computation, the messages are prepared and preprocessed to extract the different keywords "of interest". Once this preparation is operated, a ranking measure is required to select the most representative keywords of our corpus. Therefore, we use a well-known measure in the information retrieval area, *TF-IDF*. This measure is used to associate a weight to each candidate keyword in a document [2]. n our context, this weight represents the keyword relevance for the social message, *e.g.* tweet. The output is a ranked list of keywords for each message:

$$tf\text{-}idf(k_i) = \frac{tf(k_i, p)}{tf_{max}} \times \log\left(\frac{\mid T \mid}{\mid p_{p \in T \text{ and } k_i \in p} \mid}\right) \quad (1)$$

Where k_i is a candidate keyword, p a message, T the collection of tweets, $f(k_i, p)$ the frequency of k_i in p, $tf(k_i, p)$ the term frequency of k_i in p.

Then, a co-occurrence graph of keywords is created to compute the keyword similarity. Vertices denote keywords and edges denote co-occurrence relations between keywords. Co-occurrences between keywords are measured by *Dice coefficient*:

$$Dice(k_i, k_j) = \frac{2 \times p(k_i, k_j)}{p(k_i) + p(k_j)} \quad (2)$$

Where $p(k_i, k_j)$ captures the number of times the two keywords, k_i and k_j appear together in the same message.

At this stage, our approach is able to find similarities between keywords composing messages and their possible co-occurrences. The remaining step is to link messages w.r.t. their textual content. To compute this textual similarity, we define it as the average pairwise similarity ("Dice coefficient") between all the keywords of two tweets:

$$txt(p, q) = \frac{\sum_{k_i \in p} \sum_{k_j \in q} r(k_i, k_j)}{\sum_{k_i \in p} \sum_{k_j \in q} w(k_i, k_j)} \quad (3)$$

Where p and q represent two messages; $r(k_i, k_j) = w(k_i, k_j) = 2$, if the keywords are equals. The objective is to increase the similarity for tweets sharing common keywords. If the keywords are not equals, then $r(k_i, k_j) = Dice(k_i, k_j)$, and $w(k_i, k_j) = 1$.

3.1.2 Hashtag Similarity (hash)

Some words within a social message may have a special coding and can play a specific role. This is the case of hashtags in Twitter for example. Hashtags can then be also a way for users to illustrate the subject of a message. To determine the hashtag similarity of two messages, we represent in a vector the hashtags (H_p) of a tweet and then compute their "Dice" coefficient:

$$hash(p, q) = \frac{2 \mid H_p \cap H_q \mid}{\mid H_p \mid + \mid H_q \mid} \quad (4)$$

Thus, messages exhibiting the same hashtags tend to be linked and grouped together.

3.1.3 Content similarity of two messages

Following the computation of text similarities as well as hashtags similarities, we rely on these assets to compute the final similarity between the content.

$$CS(p,q) = txt(p,q) + hash(p,q) \tag{5}$$

3.2 User Similarity (US)

The considered social interaction database and the context of social networks do not consider the existing links between messages. That means that relationships between "answer" to a original message is not considered. Let u_p and u_q be users who send messages p and q respectively. Let U_p and U_q be the set of users who appear in p and q respectively, including u_p and u_q. Let $f_{u_p u_q} \in \{0,1\}$, a value capturing that there is an existing additional link like "*follows*" if the user who sends the message p follows the user who sent the message q. Let $fd_{u_p u_q} \in \{0,1\}$ be the value if the user who sends the message p is followed by the user sends the message q. We compute the user similarity using Formula 6.

$$US(p,q) = \frac{1}{2}\Big(\frac{f_{u_p u_q} + fd_{u_p u_q}}{2} + \frac{2 \mid U_p \cap U_q \mid}{\mid U_p \mid + \mid U_q \mid}\Big) \tag{6}$$

The value of this measure captures the similarity degree between the messages from users who have participated in it. Specifically, if participants are the same in both messages, the degree of similarity between these messages should then increase.

3.3 Temporal Similarity (TS)

The nature of social networks is that of a quickly and dynamically changing and evolving system and content. Thus, a piece of information having a certain interest at time t may loose it quickly at $t+1$. Thus, to materialize this quickly evolving environment, we consider the temporal dimension in the grouping process of message. To estimate the temporal similarity between two messages, it is necessary to have an upper bound of the time difference between them, *i.e.* two messages sent at far time intervals would not tend to be linked, due to the previously highlighted time property of social information. Although this hypothesis seems strong, it is justified because of the wide dynamics related to social networks. For leveraging the dynamics of social networks in the creation grouping messages, we exploit the reactivity of a person as an enabler. The right side of Figure 1 shows the underlying rationale behind our idea and its justification by confronting the phenomenon of information dissemination in these same networks.

Picking a large value will result in increasing the computation power needed for the processing. In the opposite, considering a small value would result in "ignoring" old messages, which can be of interest. To come with an objective estimation of this time, we analyzed several messages to understand the users' preferences in terms of, *e.g.* reaction time to other messages, connection times to their accounts, etc. After an evaluation in the social interactions database that we have, we recovered that a user has an average of three times connections per day. This gives a logging interval of 8 hours for each user. In the propagation of information [22], authors have shown that:

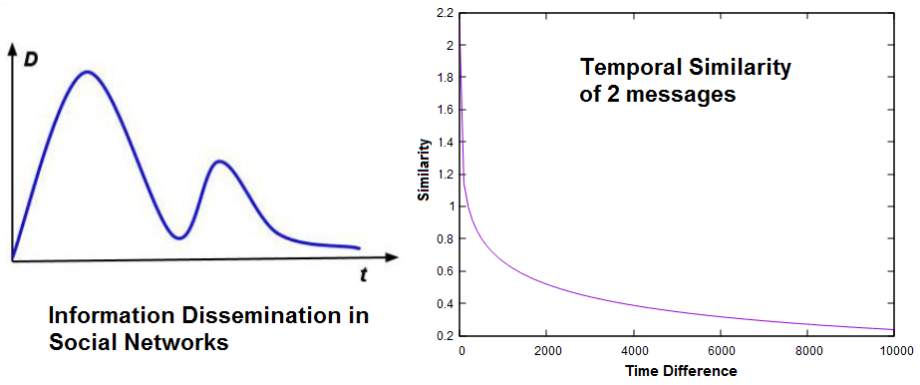

Figure 1: Illustration of (left side) the information dissemination and (right side) the effect of the time on the similarity of social messages.

(i) the propagation of information has a behavior similar to that shown at the left side of Figure 1 with a two pulse steps and (ii) users' reaction on messages within 3 hours after the launch of the discussion.

Combining these two observations, this gives us a time window of [3-8] hours. Thus, for the sake of simplification, we have decided to use an average value of 5.5 hours representing the reaction time of an user. This value can be variable depending on the needs and the desired performances. This value controls the strength or the penalty assigned to the link: the more the time passes, the more the message is becoming less important and consequently the more it goes away from the clusters.

Let d_p and d_q be the dates of each message. Let's also consider $t_w \in [3\text{-}8]$ to be the considered reference time period inside the time window ($t_w = 19800$ seconds $= 5.5$ hours in our experiments). The contribution of the time dimension to the linkage of two social messages can be formalized as:

$$TS(p,q) = \log_{100}\left(1 + \frac{t_w}{\mid d_p - d_q \mid +1}\right) \tag{7}$$

3.4 CUT: A similarity measure for short messages

Following the computation of the different dimensions, we reach to the effective linkage of messages. As highlighted before, our vision of the messages groups construction implies the consideration of the three computed dimensions, which take advantage of the inherent properties and characteristics of social networks.

$$CUT(p,q) = w_c \times CS(p,q) + w_p \times US(p,q) + w_t \times TS(p,q) \tag{8}$$

Where, w_c, w_p and w_t represent the weights given to each measure. These weights are between 0 and 1, are selected manually, and obviously $sum(w_c + w_p + w_t) = 1$.

For the effective grouping and linkage of messages, clustering algorithms can be used taking advantage of the last similarity measure. We adopt an *"Hierarchical Ascendant Clustering"* approach for clustering. The proposed method calculates a degree

of similarity between messages and existing clusters. The distance between two clusters is defined as the average pairwise distance between points in C_i and C_j is done as follows:

$$\delta(C_i, C_j) = \frac{\sum_{p \in C_i} \sum_{q \in C_j} CUT(p, q)}{\mid C_i \mid \times \mid C_j \mid} \tag{9}$$

4 Experiments and Results

To evaluate our approach, we used a data set consisting of tweets collected through Twitter's search API[1] during the period of September to October, 2012. In total, we collected $2,100,000$ tweets (excluding duplicates returned by the Twitter API). To capture the topic of the messages, hashtags are used as indicators for the linkage quality. For example, if a message contains *"#influenza"*, then the class of the message is *"influenza"*, this makes possible to identify the topic of the message.

As starting point, we selected a total of 10,000 tweets containing URI's and hashtags. Table 1 lists the top 10 hashtags of this sample as well as the predefined topics to which they belong. We can observe that these hashtags belong to the biomedical domain. More precisely, it is related to the health domain. So, as an assumption, the clustering of messages, should be performed over tweets related to health, where clusters would be formed according to a disease.

	Hashtag	***Frequency***		***Hashtag***	***Frequency***
1	#meningitis	626	6	#pathogenposse	184
2	#leukemia	290	7	#health	104
3	#hepatitis	256	8	#influenza	76
4	#measles	252	9	#vaccine	62
5	#vaxfax	184	10	#stopavn	52

Table 1: Top 10 Hashtags on our Sample.

Data preparation: Before using the messages, they have been preprocessed. The first step extracts the keywords of each message by using the *GATE* Twitter POS tagger[2] [6], an application specialized to tag tweets. Then we enrich the tweets with URI's information by using *Alchemy Api*[3]. Finally, we filter out the content of our input corpus using a list of general linguistic patterns [12]. During this step, only keywords whose syntactic structure is in the patterns list are selected, resulting in candidate keywords. We adopt to use linguistic patterns to alleviate the problem of the extraction of multi-word expressions with complex structures.

Results: For the evaluation of our clustering solutions, there exist indices that are used to measure the quality of clustering results. There are two kinds of validity indices [8]: external and internal. External indices use pre-labelled datasets with "known" cluster configurations and measure how well clustering techniques perform with respect to these known clusters. Internal indices are used to evaluate the"goodness" of a cluster configuration without any priory knowledge of the nature of the clusters.

[1] https://dev.twitter.com/

[2] https://gate.ac.uk/wiki/twitter-postagger.html

[3] http://www.alchemyapi.com/

As we mentioned before, our data set is not annotated. So, we decided to perform an evaluation with internal indices. We use for this the following measures: (i) the intra-cluser similarity average (*ISIM*), and (iii) the inter-cluster similarity average (*ESIM*). Figure 2 illustrates the results while varying the number of expected clusters. From these measures, we expect to maximize the intra-cluster values and/or to minimize the inter-clusters values, which would represent a suitable grouping. Following this intuition, we focus on the 7-ways and the 9-ways clustering.

Table 2 presents the detailed evaluation when the data set is split into 7 and 9 clusters, respectively. We can notice that the cluster number 7 of the 7-ways clustering (*Cluster-6*)) has been divided in three cluster, *i.e. Cluster-6, Cluster-7*, and *Cluster-8* of the 9-ways clustering. This division has increased the internal similarity of the new formed clusters.

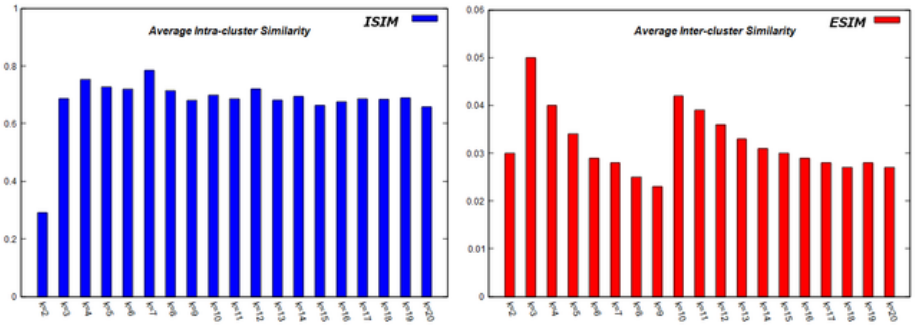

Figure 2: Average of Internal and External Similarity varying the number of k Clusters.

Id of Cluster	*Size*	*ISIM*	*ESIM*	Id of Cluster	*Size*	*ISIM*	*ESIM*
7-ways Clustering				**9-ways Clustering**			
Cluster-0	584	1.489	0.008	*Cluster-0*	584	1.489	0.008
Cluster-1	233	1.190	0.012	*Cluster-0*	584	1.489	0.008
Cluster-2	2006	0.764	0.060	*Cluster-2*	2006	0.764	0.060
Cluster-3	1237	0.741	0.089	*Cluster-3*	1237	0.741	0.089
Cluster-4	175	0.671	0.012	*Cluster-4*	175	0.671	0.012
Cluster-5	511	0.626	0.009	*Cluster-5*	511	0.626	0.009
Cluster-6	5254	0.011	0.005	*Cluster-6*	249	0.406	0.003
				Cluster-7	548	0.223	0.005
				Cluster-8	4457	0.009	0.005

Table 2: Internal Indices for k-Clusters.

5 Conclusion and future work

We have tackled in this work the problem of linking social interactions in order to reduce the information overload. We have used Twitter as an example of a social network. We have proposed an approach considering several steps and using the social network information: (i) content, (ii) users, (iii) time. The innovation in this approach

is also represented by the "massive" use of social dimension in all steps of the process ensuring a contextual linkage. The preliminary results are encouraging and show the interest of the approach.

We believe that it is necessary to solve the problem of execution time by optimizing the computation of the linkage at the different levels. A natural next step in this work is the summary of the obtained groups of messages in order to better simplify the representation for end-users and the information digestion (*i.e.* removing duplicates, keeping important information, etc). Another important issue would be a qualitative evaluation of the results by including real users. This will provide a better idea on the measurement of the information overload.

References

[1] C. C. Aggarwal and C. Zhai. A survey of text clustering algorithms. In *Mining Text Data*, pages 77–128. Springer, 2012.

[2] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., USA, 1999.

[3] A. E. Cano, A. Varga, M. Rowe, F. Ciravegna, and Y. He. Harnessing linked knowledge sources for topic classification in social media. In *Proc. of the 24th*, HT'13, pages 41–50. ACM, 2013.

[4] M. Chen, X. Jin, and D. Shen. Short text classification improved by learning multi-granularity topics. In *Proc. of the 22nd IJCAI*, IJCAI'11, 2011.

[5] Z. Dai, A. Sun, and X.-Y. Liu. Crest: Cluster-based representation enrichment for short text classification. In *Advances in KDD*, pages 256–267. Springer, 2013.

[6] L. Derczynski, A. Ritter, S. Clark, and K. Bontcheva. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proc. of the RANLP*. ACL, 2013.

[7] S. Erera and D. Carmel. Conversation detection in email systems. In *Proc. of the 31st*, ECIR'09, pages 498–505, 2008.

[8] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Cluster validity methods: part i. *ACM Sigmod Record*, 31(2):40–45, 2002.

[9] X. Hu, N. Sun, C. Zhang, and T.-S. Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proc. of the 18th*, CIKM'09, pages 919–928. ACM, 2009.

[10] W. Hua, Z. Wang, H. Wang, K. Zheng, and X. Zhou. Short text understanding through lexical-semantic analysis. ICDE, April 2015.

[11] J. A. Lossio-Ventura, H. Hacid, A. Ansiaux, and M. L. Maag. Conversations reconstruction in the social web. In *Proc. of the 21st*, WWW'12, pages 573–574. ACM, 2012.

[12] J. A. Lossio-Ventura, C. Jonquet, M. Roche, and M. Teisseire. Biomedical term extraction: overview and a new methodology. *Information Retrieval Journal*, 19(1):59–99, 2016.

[13] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proc. of the 17th*, WWWW'08, pages 91–100. ACM, 2008.

[14] A. Rangrej, S. Kulkarni, and A. V. Tendulkar. Comparative study of clustering techniques for short text documents. In *Proc. of the 20th*, WWW '11, pages 111–112. ACM, 2011.

[15] K. D. Rosa, R. Shah, B. Lin, A. Gershman, and R. Frederking. Topical clustering of tweets. 2011.

[16] L. Shou, Z. Wang, K. Chen, and G. Chen. Sumblr: Continuous summarization of evolving tweet streams. In *Proc. of the 36th*, SIGIR '13, pages 533–542. ACM, 2013.

[17] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *Proc. of the 33rd*, SIGIR'10, pages 841–842, 2010.

[18] A. Sun. Short text classification using very few words. In *Proc. of the 35th*, SIGIR'12, pages 1145–1146. ACM, 2012.

[19] G. Tang, Y. Xia, W. Wang, R. Lau, and F. Zheng. Clustering tweets using wikipedia concepts. In *Proc. of*, LREC'14, 2014.

[20] F. Wang, Z. Wang, Z. Li, and J.-R. Wen. Concept-based short text classification and ranking. In *Proc. of the 23rd*, CIKM'14, pages 1069–1078. ACM, 2014.

[21] T. Xu and D. W. Oard. Wikipedia-based topic clustering for microblogs. *Proc. of the American Society for Information Science and Technology*, 48(1):1–10, 2011.

[22] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. In *IEEE International Conference on Data Mining*. Stanford InfoLab, 2010.

[23] J. Yin and J. Wang. A dirichlet multinomial mixture model-based approach for short text clustering. In *Proc. of the 20th*, KDD '14, pages 233–242. ACM, 2014.

[24] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware point-of-interest recommendation. In *Proc. of the 36th*, SIGIR '13, pages 363–372. ACM, 2013.