IADIS International Journal on Computer Science and Information Systems
Vol. 12, No. 2, pp. 1-16
ISSN: 1646-3692

CLIQUES DETECTION vs MAXIMUM SPANNING TREE FOR TWEET CONTEXTUALIZATION

Amira Dhokar, Lobna Hlaoua and Lotfi Ben Romdhane.
SDM[1] Research Group, MARS[2] Research Lab ISITCom, University of Sousse, Tunisia

ABSTRACT

Nowadays, social medias are very popular among their users. One of the most well-known social networks is Twitter. It is a micro-blog that enables its users to send short messages called tweets. A tweet is a 140 characters long message that is rarely self-cont, hence additional information are necessary to allow better readability of the tweet. This new task has attracted a great deal of attention recently. Given a tweet, the aim of tweet contextualization is to produce an informative and coherent paragraph, called a context, from a set of documents in response to topics treated by the tweet.
In this paper, we propose a new approach of Tweet Contextualization based on combining automatic summarization techniques and sentence aggregation. The main idea of our proposed method is to select relevant, informative and semantically related sentences that best describe themes expressed by the tweet, and then build a concise context.

KEYWORDS

Tweet contextualization, cliques detection, MST

1. INTRODUCTION

The exponential growth of social networks during the last decade has affected a large diverse audience in age, culture and specialty (Perrain, 2015). It is the emergence of smart phones that facilitates this massive penetration of these networks in everyday life. In Effect, we find them present in the daily lives of their users: groups of friends circle of acquaintances, internet networks, many examples that highlight a real social phenomenon. Furthermore, the flexibility of their use makes them a very popular way of exchange among internet users.

[1] Smart Data Mining
[2] Modeling of Automated Reasoning Systems

The number of social networks continues to grow day by day, and much of the Community exchange is done through them. However social media does not have the same popularity among their users. Twitter is one of the most used social networks throughout the world (Duggan et al., 2015). It is a micro-blogging tool that provides its users the opportunity to exchange short messages, called tweets, with a maximum size that does not exceed 140 characters (Boyd et al., 2010), often in real time and from a mobile phone.

This type of message raises new issues, and the data stream generated by the tweets continues to grow. However, the limited information conveyed by such messages, because of their size, makes them sometimes not understandable. Moreover, their small size leads to the use of a specific, misspelled, or truncated vocabulary (Morchid and Linares, 2012), which provokes the fact that tweets often need to be explained. i.e., it is essential to know their original context: additional information may be necessary to allow better readability of tweets and to help users to find their needs without time consuming. Therefore providing concise and coherent context seems to be helpful.

In this article, we propose an approach for tweet contextualization based on semantic and coherence between sentences, to select the most relevant and coherent information in relation with the query and extract the most important ones to provide the appropriate context. In this respect, several questions arise: How to properly formulate a query from a tweet? How to extract the most relevant and coherent information from several sources related to one or more themes expressed by the tweet? How to build a concise and understandable context from the already selected sentences as relevant, without losing the meaning or the structure of a summary?

The remainder of this paper is organized as follows: Section 2 cites some related works. Section 3 presents our motivation and the architecture of our model. Section 4 discusses relevant sentences extraction from a document. Section 5 details the proposed approach to build an informative and a coherent context using sentences aggregation. Section 6 describes our experiment results. The conclusion and future work are presented in Section 7.

2. RELATED WORKS

Tweet contextualization is a new issue that aims to answer questions of the form “What is this tweet about?” which can be answered by selecting several sentences from different documents. Thus, the general process of a tweet contextualization system involves tweet analysis, XML/passage retrieval and automatic summarization to get a response to this question. Though the idea is quite recent, it knows a craze within the scientific community. Indeed, it is treated by several approaches which we categorized into three classes.

2.1 Based-Retrieval System and Summarization for Tweet Contextualization

Several studies have found that combining an Information Retrieval system (IR) and a summarization system perfectly resolves the conflict generated by the tweet contextualization. Deveaud and Boudain (2013) used an approach composed of three main components: preprocessing, Wikipedia articles retrieval and multi-document summarization. They segment the most important articles related to the tweet, into sentences and compute a score for each

phrase regarding their importance to the Tweet, URLs embedded in the Tweet, and the centrality within the article from which the sentence is extracted. The most important sentences are then selected for the context. Ermakova and Mothe (2013) proposed a new method for tweet contextualization based on TF-IDF cosine similarity measure enriched by smoothing from local context, named entity recognition and part-of-speech weighting. After selecting a define number of relevant sentences from relevant documents, they aim to reorder them by modeling the task as sequential ordering problem. They used a greedy algorithm to solve the sequential ordering problem, where vertices corresponded to sentences and sequential constraints were represented by sentence time stamps. Linhares (2013) used an automatic greedy summarizer named REG, which uses a greedy optimization algorithm to weigh the sentences. The context is obtained by concatenating relevant phrases weighed in the optimization step.

2.2 Based-Query System for Tweet Contextualization

The second family considers that the query (tweet) is the most important part of a tweet contextualization system. One of the most known techniques in this context is proposed by Morchid and Linares (2012). They used latent Dirichlet analysis (LDA) for a tweet thematic representation. It allows them to extend the tweet vocabulary by a set of thematically closed words, to obtain a more robust query. Another approach is used by Zingla et al. (2014) to allow the extension of the tweet's vocabulary by a set of thematically related words using mining association rules between terms. After generating the association rules using an efficient algorithm, authors proposed to project the tweets on the set of the association rules in order to obtain the thematic space of each tweet. The last step of the proposed method is to Sending the query to the baseline system, composed of an Information Retrieval System (IRS) and an Automatic Summary System (ASS), to extract from a provided Wikipedia corpus a set of sentences representing the tweet context that should not exceed 500 words.

2.3 Based-Retrieval System for Tweet Contextualization

Another line of research is based on the fact that a retrieval system can be suitable for tweet contextualization. Bhaskar et al. (2012) consider the tweet contextualization task as a passage retrieval task. After some reprocessing, each tweet was submitted, as a query, to the search engine and the obtained passages formed the desired context in response to the initial query. The method proposed by Ganguly et al. (2012) involves indexing the passages in Wikipedia articles as separate retrievable units, extracting sentences from the top ranked passages, and then select the top most similar ones with respect to the query.

Systems that combine a retrieval system and summarization techniques performed well on Tweet Contextualization (Bellot et al., 2013). They had encouraging results in relevance and precision since they consider informative and coherent sentences to form the context. However, approaches relayed on query expansion give encouraging results on how well the summary explains the tweet, but do not perform very well on coherence. In an other hand, considering a tweet contextualization system as a retrieval one didn't obtain good performances in this task. Indeed, this type of systems suffers from too much of noise both in relevance and coherence.

Despite the attempt of various methods to overcome the tweet contextualization, this problem still remains. Inspired by the above approaches, in this paper, a semantic approach for tweet contextualization based on sentence aggregation is proposed.

3. MOTIVATION AND ARCHITECTURE OF THE MODEL

Our objective is to select most relevant and coherent parts of documents correlated to an initial query to be able to construct an appropriate context. In view of importance of both relevance and coherence in a contextualization system, we have considered that sentences extraction is a very important step in a tweet contextualization system. Hence our approach consists of extracting important, relevant, and semantic related sentences from a set of documents to construct a context that describes as best as possible themes treated by the tweet. After extracting relevant sentences, we proposed an aggregation sentences step to select the most relevant and coherent sentences according to a criteria of importance. Hence, we consider that a candidate sentence should be relevant regarding the Tweet, informative in the document from which the phrase is selected and Coherent with other sentences.

Our method aims to filter candidate sentences and select most relevant ones related to the tweet. Theses selected sentences (regarding the Tweet), have to be also relevant in the corresponding document: We first consider that the title of a given document can best describes the article. So, we consider that a relevant sentence should be in relation with the title's document. On an other hand, we consider a sentence in a document as informative if it is highly correlated to other sentences in the same document. Furthermore, a selected sentence had to be semantically related with other phrases to ensure higher degree of readability and coherence to the constructed context.

We opted to combine these aspects to extract the most relevant and semantically related parts of relevant documents. The main idea is to select the most relevant and coherent phrases according to a criteria of importance and to build an excerpt. This approach will be detailed in the next section. The aim of a tweet contextualization system is to provide a context that allows better readability for a given tweet. To enhance the quality of this context, i.e., ensuring that the context summaries contain adequate correlating information with the tweets and avoiding the inclusion of relevant and coherent information, we proposed an approach based on sentence aggregation from many documents. The general process of tweet contextualization involves three steps: Tweet Analysis, passage and/or XML elements retrieval and construction of the answer (context).

Based on this process, our proposed approach for tweet contextualization is depicted in Figure 1. It is mainly composed of the following three steps:

1) Tweet analysis: We used this step to clean tweets and eliminate unnecessary symbols such as #, @...and URLs.
2) Passages/XML documents retrieval: The cleaned query is transmitted to the search engine, to determine the most relevant articles to the tweet.
3) Tweet contextualization: This part of our proposed contextualization system is the most important part of our model. It aims to extract then reorder most relevant and coherent sentences related to the tweet. Top-ranked phrases are selected to form context (within the limit of 500 words).

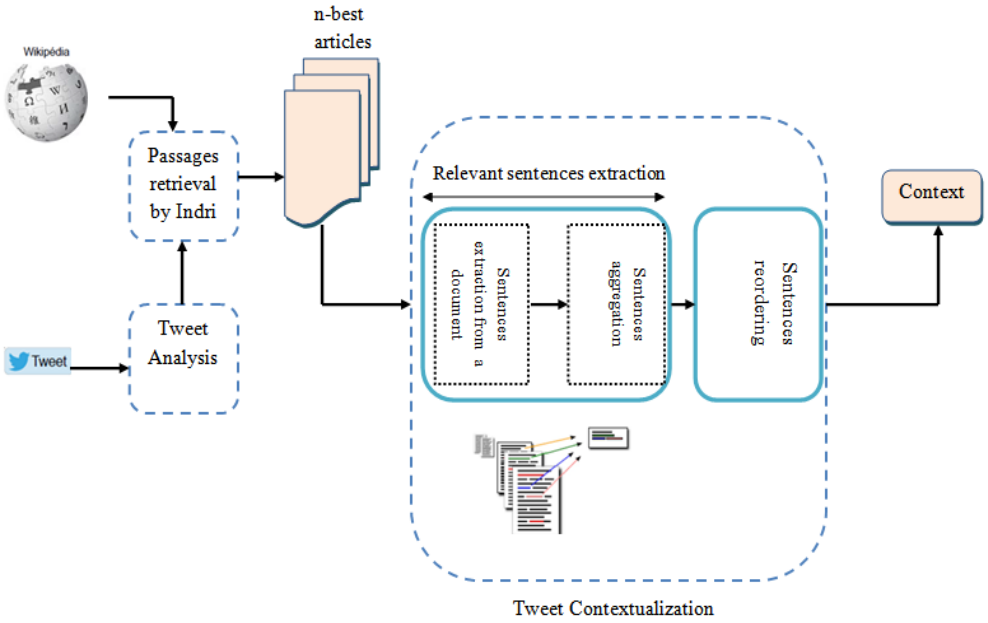

Figure 1. The proposed contextualization model

4. RELEVANT SENTENCES EXTRACTION FROM A DOCUMENT

Retrieving relevant sentences for each relevant document is the first crucial part of our system. This step is performed through the following two steps: Document filtering regarding the tweet and sentence scoring.

4.1 Document Filtering Regarding the Tweet

The aim of this step is to select the most informative phrases related to the topics treated by the tweet. Hence, we filter the corresponding document by keeping only sentences that are correlated to the query by calculating the cosine similarity between the tweet and the candidate sentences given by:

$$\text{Similarity}(Q,S) = \frac{\sum_{i=1}^{n} \text{Freq}(q_i,Q)}{\sqrt{\sum_{i=1}^{n} (\text{Freq}(q_i,Q))^2}} \times \frac{\sum_{i=1}^{n} \text{Freq}(s_i,S)}{\sqrt{\sum_{i=1}^{n} (\text{Freq}(s_i,S))^2}} \qquad (1)$$

Where $Q = q_1, q_2, \ldots, q_i$ is a query

$S = s_1, s_2, \ldots, s_i$ is a sentence

$\text{Freq}(q_i, Q)$ is the occurrence of the i-th token in a query

$\text{Freq}(s_i, S)$ is the occurrence of the i-th token in a sentence.

If the token is not presented in the query or in the sentence, q_i (resp.s_i) is equal to 0 respectively.

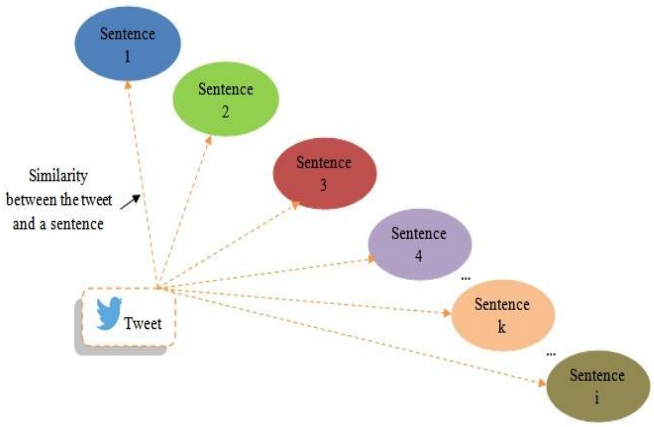

Figure 2. Document filtering regarding the tweet

4.2 Sentence Scoring

We have to select the most relevant sentences in the filtered document. For each candidate sentence, a score is computed to evaluate the importance of a phrase in the corresponding document. This score is based on:

- The relevance of the sentence compared to the title of the document.
- The importance of the sentence in its original document compared to other sentences in the same article.

The best scored sentences are selected. The score of each phrase is given by:

$$Sp_i = Similarity(T, S_i) + Imp(S_i) \qquad (2)$$

Where Sp_i is the associated score of a sentenceS_i, Similarity (T, S_i) is the similarity estimated between a sentence S_i and the title of the document T and Imp (Si) is the score that estimates the importance of a sentence S_i in a document. Features used in (2) are treated more in details in the following.

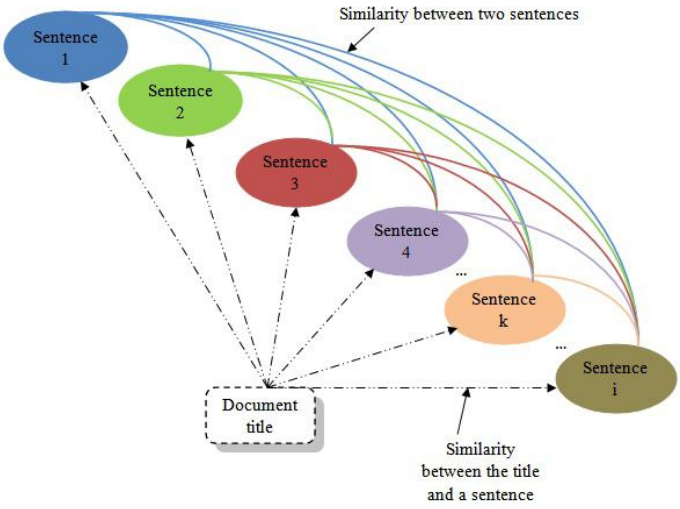

Figure 3. Parameters used to calculate the score of a sentence

4.2.1 Sentence Relevance with Respect to the Title

The idea of considering that the title is the best element that can summarize a text is not new. In fact, several studies have adopted this idea and had promising results (Edmundson, 1969). We therefore hypothesize that the sentences that are closer to the title are intuitively the most relevant in a document. For this, we calculate the cosine similarity between each candidate sentence and the title of the document expressed by the following equation:

$$\text{Similarity}(Q,S) = \frac{\sum_{i=1}^{n} \text{Freq}(t_i,T)}{\sqrt{\sum_{i=1}^{n}(\text{Freq}(t_i,T))^2}} \times \frac{\sum_{i=1}^{n} \text{Freq}(s_i,S)}{\sqrt{\sum_{i=1}^{n}(\text{Freq}(s_i,S))^2}} \quad (3)$$

Where T=$t_1,t_2,\ldots,t_i$ is the title of the corresponding document.

S=$s_1,s_2,..,s_i$ is a sentence.

Freq(t_i,T) is the occurrence of the i-th token in a title.

Freq(s_i,S) is the occurrence of the i-th token in a sentence.

If the token is not presented in the title or in the sentence, q_i (resp.s_i) is equal to 0 respectively.

4.2.2 Sentence Importance in a Document

Sentences do not have the same importance within a document. In each article, we have to extract the most informativeness ones. i.e., sentences that contain more information compared to other ones in the same document. Hence, we calculate a score for each candidate sentence in the document. It is calculated until divergence and given by (Brin and Page, 2012):

$$Imp(S_i) = (1-d) + d \times \sum_{S_j \in Neighbors\,(S_i)} \frac{Sim(S_i,S_j)}{\sum_{S_k \in Neighbors\,(S_i)} Sim(S_k,S_i)} \times Imp(S_i) \quad (4)$$

Where d is a dumping factor (usually set d to 0.85), Neighbors (S_i) is the set of sentences connected with S_i and Sim (S_i,S_j) is the similarity score between sentences S_i and S_j and given by (Mihalcea, 2004):

$$Sim(S_i, S_j) = \frac{\sum_{m \in S_i, S_j} freq(m, S_i) + freq(m, S_j)}{\log|S_i| + \log|S_j|} \quad (5)$$

Where, freq (m,S_i) is the occurrence of a word m in a sentence S_i, respectively S_j and log |S_i| is the length of a sentence S_i , respectively S_j.

5. SENTENCE AGGREGATION

5.1 Motivation

As we mentioned in the previous section, we work with the n top documents from the research phase. From each document, we select best scored sentences and aggregate them together. However, a good context should have a good quality. i.e., sentences should be correlated to each other to ensure coherence between phrases. In this respect, we propose to use two methods for sentences aggregation: sentences aggregation using cliques detection and sentences aggregation using maximum spanning tree.

Figure 4. Best scored sentences aggregation

5.2 Sentence Aggregation Based on Cliques Detection (SACD)

Identifying cliques can help find cohesive subgroups in a graph. Usually, each node in a clique is, in some way, highly related to every other node (see Figure. 5). This characteristic makes clique identification a very important approach to uncover meaningful groups of sentences from a graph. In this section, we opted for finding maximal cliques of a graph to identify coherent sentences in order to produce a coherent context.

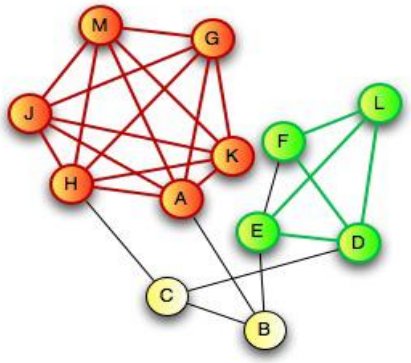

Figure 5. An example of cliques detection

5.2.1 Text as a Graph: Sentence Similarity Graph

Salton et al. (1997) and Yeh et al. (2008) employed techniques for inter-document link generation to produce intra-document links between passages of a document, and obtained a text relationship map (or content similarity graph). Inspired by these works, we try to adopt the same idea and to model a group of sentences (here aggregated sentences resulted from the first step of our system) as a network of sentences that are related to each other, resulting in a sentence similarity graph. A sentence similarity graph is defined as a graph with nodes and edges linking nodes. Each node in the graph stands for a sentence (Yeh et al., 2008). Two sentences are connected if and only if they are similar with respect to a similarity threshold α. i.e., an edge between two nodes indicates that the corresponding two sentences are considered to be semantically related. The degree of similarity between two sentences S_i and S_j is measured by the formula proposed in 5.

5.2.2 Cliques Computation

Our approach to identify cliques is based on the notion of a maximal clique. A maximal clique of a graph G is a clique that cannot be extended by including one more adjacent vertex(Regneri, 2007). Cliques are allowed to overlap, which means that sentences can be members of more than one clique. We consider an undirected graph G= (V, E). We let n and m be the number of vertices and edges of G, respectively. For a vertex v, we define $\tau(\vartheta)$ to be the set $\{\omega|(\vartheta, \omega) \in E\}$ which we call the neighborhood of v.

The purpose of this step of our work is to detect all maximal cliques present in the graph using Tomita algorithm (Tomita et al., 2011), developed by Tomita et al. This algorithm is a modified version of the Bron-Kerbosch algorithm (Bron and Kerbosh, 1973). The aim of using such algorithm is to generate all maximal cliques, so that we can detect coherent and related groups of sentences.

A recursive call to the Bron-Kerbosch algorithm provides three disjoint sets of vertices R, P, and X as arguments, where R is a (possibly non-maximal) clique and $P \cup X = \tau(R)$ are the vertices that are adjacent to every vertex in R. The vertices in P will be considered to be added to clique R, while those in X must be excluded from the clique; thus, within the recursive call, the algorithm lists all cliques in $P \cup R$ that are maximal within the sub graph induced by $P \cup R \cup$X. The algorithm chooses a candidate v in P to add to the clique R, and makes a recursive

call in which v has been moved from R to P; in this recursive call, it restricts X to the neighbors of v, since non neighbors cannot affect the maximality of the resulting cliques. When the recursive call returns, v is moved to X to eliminate redundant work by further calls to the algorithm. When the recursion reaches a level at which P and X are empty, R is a maximal clique and is reported. To list all maximal cliques in the graph, this recursive algorithm is called with P equal to the set of all vertices in the graph and with R and X empty(Eppstein et al., 2010). Tomita algorithm uses a specific pivoting policy to cut computational branches. The pivoting consists in the following: instead of iterating at each expansion on the P set, chose a pivot. The results will have to contain either the pivot or one of its non-neighbors, since if none of the non-neighbors of the pivot is included, then we can add the pivot itself to the result. Hence we can avoid iterating on the neighbors of the pivot at this step (they will still be expanded in the inner levels of recursion). The strategy proposed by Tomita et al. is to choose the pivot as the node in $P \cup X$ with the highest number of neighbors in P (Tomita et al., 2011).

Algorithm 1 Maximal cliques detection with pivot

if $P \cup X = \emptyset$ **then**
  report R as a maximal clique
end if
choose a pivot $u \in P \cup X$ {Tomita et al. choose u to maximize$|P \cap \tau(u)|$ }
for each vertex $v \in P \backslash \tau(u)$ **do**
  Tomita$(P \cap \tau(v), R \cup \{v\}, X \cap \tau(v))$
  $P \leftarrow P\{v\}$
  $X \leftarrow X \cup \{v\}$
end for

The following figures from Tomita et al. (2006) illustrate an example of maximal cliques detection.

Figure 6 represent an input graph, figure 7 is a search forest for the graph G, and figure 8 shows a resulted printed sequence.

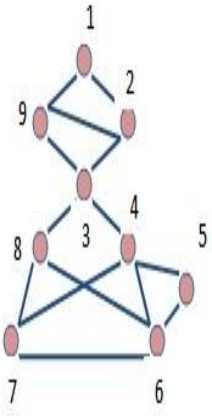

Figure 6. An input graph *G*

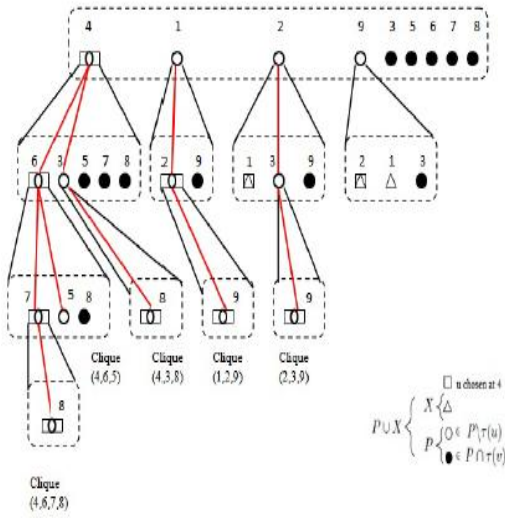

Figure 7. A search forest for G

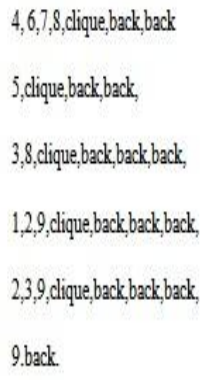

Figure 8. A resulting sequence

5.3 Sentence Aggregation Based on Maximum Spanning Tree (SAMST)

The use of trees started in the middle of the 19th century. The simplicity of the tree representation makes it a method of choice today to easily convey the diversification and relationships between sentences.

We opted for this method not only for selecting coherent and semantically related phrases, but also to make an order between sentences that construct the maximum spanning tree.

5.3.1 From a Text to a Tree

We used the same principle of the previous section. i.e, a sentence similarity graph is defined. Nodes correspond to sentences and edges represent the weights between nodes. Two sentences are connected if and only if they are similar with respect to a similarity threshold α. i.e., an edge between two nodes indicates that the corresponding two sentences are considered to be semantically related. The degree of similarity between two sentences S_i and S_j is measured by the formula proposed in 5.

5.3.2 Maximum Spanning Tree Computation

Our proposed method consists on identifying a group of nodes of the tree (here a group of sentences) coming from many documents that are semantically related. Our approach is based on maximum spanning tree. Given a connected and undirected graph, a spanning tree of that graph is a subgraph that is a tree and connects all the vertices together. A single graph can have many different spanning trees. We can also assign a weight to each edge, which is a number representing how unfavorable it is, and use this to assign a weight to a spanning tree by computing the sum of the weights of the edges in that spanning tree. A maximum spanning

tree (MST) or maximum weight spanning tree is then a spanning tree with weight larger than or equal to the weight of every other spanning tree. More generally, any undirected graph (not necessarily connected) has a maximum spanning forest, which is a union of maximum spanning trees for its connected components. The purpose of this step is to detect a maximum spanning tree using Prim's algorithm (Prim, 1957).This algorithm is usually used to find a minimum spanning tree for a connected weighted graph. If we inverse all weights of the adjacency matrix, finding a maximum spanning tree of the original graph is equivalent to find the minimum spanning tree of the graph with new weights. So Prim's algorithm can be used to find a maximum spanning tree. The principle of this algorithm is to finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is maximized. The algorithm continuously increases the size of a tree, one edge at a time, starting with a tree consisting of a single vertex (in our case, we propose to begin with the vertex (sentence) with the highest score calculated in the previous step), until it spans all vertices. We considered the order for nodes (sentences) in the detected maximum spanning tree to produce an informative and coherent context (within the limit of 500 words). Figure. 9 illustrate an example of maximum spanning tree detection using Prim's algorithm, from an input graph to an output graph.

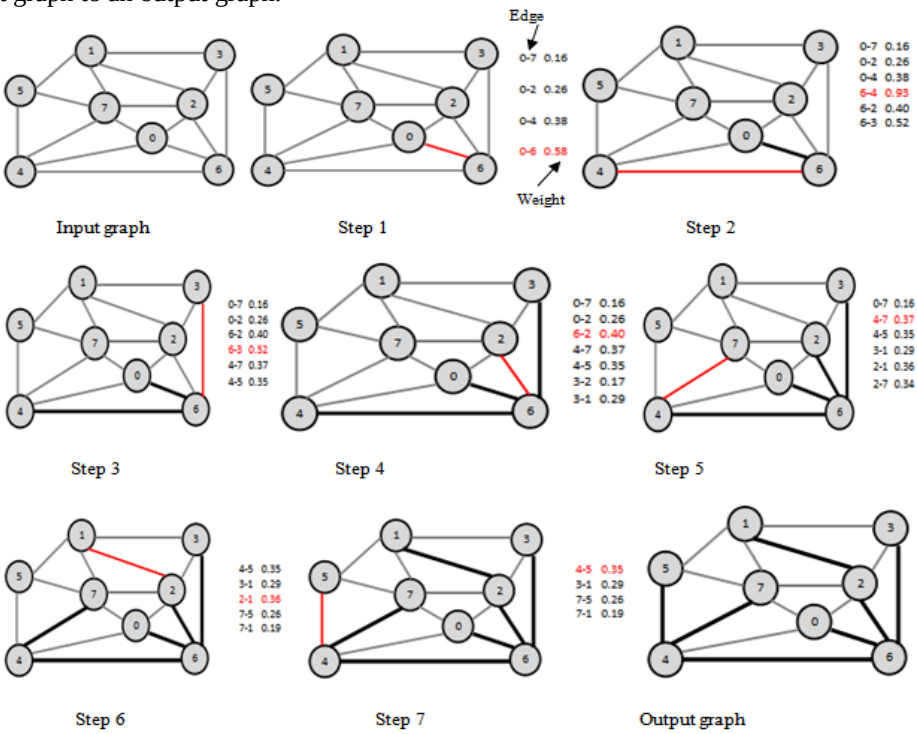

Figure 9. An example of maximum spanning tree detection using Prim's algorithm

6. EXPERIMENTS, RESULTS AND DISSCUSSION

The main purpose of this section is to analyze preliminary results given by our contextualization system. We compared our proposed method with results provided by INEX (Initiative for the Evaluation of XML Retrieval). Before reporting the experimental results, we need to indicate the Test Data and evaluation criteria that we will consider.

6.1 Description of the Test Data

To evaluate our experiments, we use the collection of articles and tweets provided by INEX. The document collection has been built based on a recent dump of the English Wikipedia from November 2012, composed of 3 902 346 articles, all notes and bibliographic references that are difficult to handle are removed and only non-empty Wikipedia pages (pages having at least one section) are kept. 70 tweets were considered for evaluation.

6.2 Evaluation Measures

Contexts are evaluated according to readability and informativeness (Bellot et al., 2013). Readability aims at measuring how clear and easy it is to understand the summary and is manually evaluated. However, informativeness aims at measuring how well the summary explains the tweet or how well the summary helps a user to understand the tweet content. This measure is based on lexical overlap between a pool of relevant passages and participant contexts (SanJuan et al., 2012). It's calculated by:

$$Dis(T,S) = \sum_{t \in T}(P-1) \times \left(1 - \frac{\min((\log(P),\log(Q))}{\max((\log(P),\log(Q))}\right) \quad (6)$$

Where $P = \frac{f_T(t)}{f_T} + 1, \quad Q = \frac{f_S(t)}{f_S} + 1$

T is a set of query terms present in reference summary and for each $t \in T$, $f_T(t)$, the frequency of term t in reference summary, S, a set of query terms present in a submitted summary and for each $t \in S$ and $f_S(t)$, the frequency of term t in a submitted summary. T may takes three forms:

- Unigrams made of single lemmas.
- Bigrams made of pairs of consecutive lemmas (in the same sentence).
- Bigrams with 2-gaps also made of pairs of consecutive lemmas but allowing the insertion between them of a maximum of two lemmas.

6.3 Results and Discussion

This section highlights experimental results of our proposed method. Unfortunately, readability is manually evaluated. Hence we can evaluate only informativeness for tweet contextualization. For this, we conducted three simulations, namely:

- ***run-1***: In this run we consider only the first part of our tweet contextualization system, i.e., we evaluated the performance of our system based on sentences extraction from only one document.
- ***run-SACD***: In this run we combine results from many documents using SACD method proposed in the previous section.

- ***run-SAMST***: In this run we also combine results from many documents using SAMST method also proposed in the previous section.

We have compared our runs with the following different runs submitted by INEX 2013 participants:

- In ***Best run***, participants (Devaud and Boudain, 2013) used hashtag preprocessing. They also used all available tweet features including web links.
- In ***REG run***, participants (Linhares, 2013) used an automatic greedy summarizer named REG (REsumeur Glouton) which uses graph methods to spot the most important sentences in the document.

6.3.1 System Performance Considering Only Sentence Extraction Step

We consider contexts formed with sentences coming from only one document. Table 1 shows informativeness results for our proposed method considering only one document. Regarding this evaluation metric, we observed that our results suffer from too much noise and need to be cleaned. This can be explained by the fact that forming a context from only one document can neglect important sentences in other relevant documents.

Table 1. Table of informativeness results considering only sentence extraction step

Run	Unigram	Bigram	Bigram with 2-gaps
Best run	0.7820	0.8810	0.8861
Run-1	**0.9420**	**0.9697**	**0.9769**
REG run	0.8793	0.9781	0.9789

6.3.2 System Performance using SACD method

Table 2 shows informativeness results for our proposed method based on cliques detection from two documents. We have considered that two sentences are connected if and only if they are similar with respect to a similarity threshold $\alpha >= 0.5$. We choose to form the context with 2, 3 and 4 cliques. We can note that our proposed approach gives courageous informativeness result compared to other systems proposed at INEX 2014. This can be explained by the fact that using cliques for sentences selection offers interesting results and helps ensure that contexts contain adequate correlating information with the evaluated tweets. The use of cliques detection improved our results by decreasing the dissimilarity between the Bigrams with 2-gaps included in the submitted summary and those included in the reference summary (0.97 vs 0.96).

Table 2. Table of informativeness results using SACD method

Run	Unigram	Bigram	Bigram with 2-gaps
Best run	0.7820	0.8810	0.8861
Run-1	0.9420	0.9697	0.9769
run-SACD	**0.8586**	**0.9220**	**0.9672**
REG run	0.8793	0.9781	0.9789

Noted that there is a real improvement of the evaluation metric compared to run-1, we try to adjust the similarity threshold α. For that, we conducted many simulations reported in table 3. We noted that there is an interesting improvement for the informativeness result with a threshold equal to 0.6 compared with the initial threshold we start with, which is 0.5.

Table 3. Table of informativeness results with different threshold

Threshold	Unigram	Bigram	Bigram with 2-gaps
0.30	0.8731	0.9832	0.9840
0.35	0.8793	0.9781	0.9786
0.40	0.8539	0.9700	0.9712
0.45	0.8643	0.9677	0.9697
0.50	0.8586	0.9220	0.9672
0.55	0.8995	0.9592	0.9620
0.60	**0.8673**	**0.9540**	**0.9575**
0.65	0.8643	0.9677	0.9680
0.70	0.8817	0.9767	0.9772

6.3.3 System Performance using SAMST Method

Table 4 shows informativeness results for our proposed method based on sentences aggregation using MST algorithm.

Table 4. Table of informativeness results using SAMST method

Run	Unigram	Bigram	Bigram with 2-gaps
Best run	0.7820	0.8810	0.8861
run-SAMST	**0.8572**	**0.9600**	**0.9672**
REG run	0.8793	0.9781	0.9789

We can note that our method using MST algorithm gives encouraging informativeness result. MST method can ensure selecting correlated and ordered sentences, but still suffer from too much of noise compared with our method using cliques detection.

To recapitulate, we can say that our system using SACD method performs better then the system using SAMST. It is also interesting to note that adjusting the threshold was beneficial, and confirmed that the proposed method using cliques detection can ensure selecting informative and correlated sentences. SACD method seems to be appropriated to the concept of tweet contextualization.

7. CONCLUSION AND FUTURE WORK

In this paper, we propose two different approaches for tweet contextualization: SACD and SAMST. The experimental study was conducted on INEX 2013 collections. The obtained results for the two methods are courageous. Results for the proposed methods are compared: The obtained results confirm that using cliques can ensure selecting coherent sentences. Indeed, experimental results through the different performed runs with different thresholds showed a satisfactory improvement in the informativeness of the contexts. We propose in our future work to develop a method for cliques selection. We propose also to work with three or more documents for sentences aggregation and to make an order for phrases in the context to improve the quality of the context with respect to informativeness and readability.

REFERENCES

Bellot, P. et al., 2013, *Overview of inex tweet contextualization 2013 track*. in CLEF.

Bhaskar, P. et al., 2012, *A hybrid tweet contextualization system using ir and summarization*," in INEX, p. 164.

Boyd, D. et al., 2010, *Tweet, tweet, retweet: Conversational aspects of retweeting on twitter*. System Sciences (HICSS). 43rd Hawaii International Conference on. IEEE, pp. 1–10.

Brin, S. and Page, L.2012, *Reprint of: The anatomy of a large-scale hypertextual web search engine*, Computer networks, vol. 56, no. 18, pp. 3825–3833, 2012.

Bron, C. and Kerbosch, J., 1973, *Algorithm 457: finding all cliques of an undirected graph*, Communications of the ACM, vol. 16, no. 9, pp. 575–577.

Deveaud, R. and Boudin, F., 2013, *Effective tweet contextualization with hashtags performance prediction and multi-document summarization*, In INitiative for the Evaluation of XML Retrieval (INEX), 2013, pp. n–a.

Duggan, M. et al., 2015, *Social media update 2014*. Pew Research Center, vol. 19.

Edmundson, H. P., 1969, *New methods in automatic extracting*, Journal of the ACM (JACM), vol. 16, no. 2, pp. 264–285.

Eppstein, D. et al., 2010, M., *Listing all maximal cliques in sparse graphs in near-optimal time*. Springer, 2010.

Ermakova, L. and Mothe, J., 2013, *Irit at inex 2013: Tweet contextualization track*, In INitiative for the Evaluation of XML Retrieval (INEX).

Ganguly, D. et al. 2012, *Dcu@ inex-2012: Exploring sentence retrieval for tweet contextualization.*

Linhares, A. C., 2013, *An automatic greedy summarization system at inex 2013 tweet contextualization track*. In CLEF (Working Notes). Citeseer.

Mihalcea, R., 2004, *Graph-based ranking algorithms for sentence extraction, applied to text summarization*, in Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. Association for Computational Linguistics p. 20.

Morchid, M. and Linares, G., 2012, Inex *2012 benchmark a semantic space for tweets contextualization*, INEX, vol. 2012. Citeseer, 2012, p. 203.

Perrin, A., 2015, *Social media usage: 2005-2015*.

Prim, R.C., 1957, *Shortest connection networks and some generalizations*, Bell System Technical Journal, The vol. 36, 1 ,p. 1389-1401.

Regneri, M., 2007, *Finding all cliques of an undirected graph*, in Seminar- Current Trends in IE WS Jun.

Salton, G. et al., 1997, *Automatic text structuring and summarization*, Information Processing & Management, vol. 33, no. 2, pp. 193–207.

SanJuan, E. et al., 2012, *Overview of the inex 2012 tweet contextualization track*," Initiative for XML Retrieval INEX, p. 148.

Tomita, E. et al., 2006, *The worst-case time complexity for generating all maximal cliques and computational experiments*, Theoretical Computer Science, vol. 363, no. 1, pp. 28–42.

Tomita, E. et al, 2011, *Efficient algorithms for finding maximum and maximal cliques: Effective tools for bioinformatics*. INTECH Open Access Publisher.

Yeh, J. Y., 2008, *ispreadrank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network*, Expert Systems with Applications, vol. 35, no. 3, pp. 1451–1462.

Zingla, M. et al., 2014, *Inex2014: Tweet contextualization using association rules between terms*, in CLEF (Working Notes), pp. 574–584.