A Topic Recommendation System for Twitter

Data mining and clustering techniques for topic extraction and user profiling

Annalisa Congiu
189264
annalisa.congiu†

Maria Pia Natale
189160
mariapia.natale†

Alessia Tovo
189192
alessia.tovo†

† @studenti.unitn.it

ABSTRACT

Nowadays, social media, like Twitter, are services more and more used by common and famous people to communicate thoughts, share videos, images and links on the Web. Retrieving useful and interesting information from what users post on Twitter is the general way to obtain a recommendation system and a user profiling system that allow the social media to monitor its users and give them a better service quality. Extracting topics from tweets text, organising them to have association rules and the application of the k-means algorithm are the main steps executed to give a solution to the stated problem.

Keywords

Data mining, Twitter, topic extraction, k-means, clustering, user profile

1. INTRODUCTION

Topic detection consists of identifying a word or a short list of words that allows to indicate a subject or concept of a document. Topic detection is an important study in many fields, such as data mining, natural language processing, sentiment classification and information retrieval.
This work is focused on the data mining approach on Twitter contents. In particular, the goal of this project is to identify a topic from each tweet and correlate tweets and users. This creates a simple recommendation system useful for users on Twitter and companies that would like to take advantage of these notions to create tailored advertisement and have a better impact on users.

Twitter represents one of the most important and most used social network in the world, by both normal users and famous people, such as politicians, writers, actors, etc. Users on Twitter may post a tweet that consists of texts, photos or videos and each user can comment, re-tweet or like other users' posts. Using Twitter's data is possible to identify groups of users that have similar interests, depending on what they talk about and, consequently, suggest new users to follow. Moreover, it is possible to recommend new topics to read about that may attract the interest of the user.

Identification of topics on tweets is more complex compared to longer documents, such as papers or manuals, due to their brevity and the noise they contain, for instance the frequent use of slang. For this reason, most of the work has been done with the aim to obtain meaningful sentences to analyse.
To extract topics from the sentences two different approaches were used: *Latent Dirichlet Allocation* (LDA) and *hashtag extraction*. Results obtained from both approaches were put together in order to have more reliable results.
The second step of the work consisted of the retrieval of the frequent itemsets in order to find "similar" topics. In this case, the similarity between topics does not refer to a semantic similarity among words but it depends on the frequency of the topics treated by users.

At this point, a set of association rules has been computed in order to suggest to users the topics they might be interested in. These rules were based on what users tweet about and on the frequent itemsets retrieved in the previous step.

To provide additional information about users and topics on Twitter, a clustering approach has been adopted to divide topics into different groups. For this purpose, the k-means clustering algorithm has been chosen because it creates a certain number of clusters and a list of tweets and users that belong to each cluster. A *TF-IDF matrix* has been created in order to merge all the texts in a readable way for the clustering algorithm.
This type of recommendation is more general than the previous one, since each cluster contains a huge number of elements while the frequent itemsets are more restricted.
Finally, the results were analysed in order to understand these correlations and to create a new user profile with the topics already in his or her tweets and to recommended topics on which he or she could be interested.

The entire work has been done on a dataset of only English tweets using Python and some specific libraries to manage the data and obtain the results.

2. RELATED WORK

Managing noisy and short text like the tweets or finding

DOI: 10.1145/1235

a way to retrieve topics from texts represent huge problems in data mining. Indeed, several data analysts worked on this field in order to find better solutions. For instance, the work proposed by Weng, Lim, Jiang and He in [1] focuses on identifying influential users in blogging services like Twitter. They state that the service already offered by Twitter to identify the users that have a lot of influence, on the base of the number of their *followers*, is not so accurate, since the "*following relationship*" is casual most of the times. On the other hand, the "following relationship" may indicate a very strong similarity between users. For this reason, the authors proposed a new model that considers the content of the tweet and subsequently applies their ranking algorithm, based on PageRank, to the data extracted.

Moreover, the *Short and Sparse Text Topic Modelling via Self Aggregation*[2] paper written by Quan, Kit, Ge, Pan refers to an approach based on the extraction of topics from short texts on social media particularly focusing on the sparsity problem of the data. The authors presented a self-aggregation based topic model, the *SATM*, for this type of texts by a natural integration of clustering and topic modelling. The model follows one of the standard topic models, the LDA, with the addition of a second phase that generates few short text snippet, which corresponds to the assumption that each text snippet is sampled from a long document. The overall results are better than the classical approach of the LDA method.

Whereas, the framework proposed by Wang et al. in [3] for topic identification in noisy and short text is based on two main steps. The first step is the feature selection of the text; then the message content and the correlations between messages are modelled. As feature selection the lasso has been used, thus the penalisation of the $l_1 - norm$. For the modelling of the message content, the Unigram model, the Term Frequency (TF) model and the term frequency-inverse document frequency model (TF-IDF) model have been taken into account. To model the social context, the authors used two social theories, one based on *Reference consistency regularisation* and the other on *Social contagion regularisation*

Another problem the data analysts have to face to is the clustering problem and which is the best algorithm to apply. In [4] the authors combine an PSO algorithm and the k-means algorithm to cluster documents. In this work the multi-dimensional document vector space is modelled as a problem space in order to apply the hybrid-algorithm. Each term in the document is represented by one dimension of the problem space whereas each document vector is a dot in the space. Then the *PSO-K-means* algorithm is executed on the transformed data. In the first stage, the PSO part is executed for a short period to search cluster centroid locations. Then, the location are used in the second stage by k-means for refining and generating the final optimal solution.

3. PROBLEM STATEMENT

To describe the problem that the solution proposed in this report aims to solve, it is necessary to define in a more formal way the given problem and the specific concepts we met during its resolution. Our problem can be subdivided in two main parts: one is to extract and correlate *topics* from Twitter text data; the other is to delineate a procedure to create *profiles* for Twitter users. We start by giving a formal definition of the concepts of *topic* and *user profile*.

Definition 3.1 (Topic)**.** The term topic is defined as a set $T = \{w_1, w_2, ..., w_n\}$ of n words that are associated together with a general argument that can represent each of the n words.

Definition 3.2 (User profile)**.** Given a set of topics $T = \{t_1, t_2, ..., t_n\}$ and a user u from the set of users $U = \{u_1, u_2, ..., u_m\}$, where n is the number of topics and m the number of users, a user profiling is an association $up : (t_1, .., t_k) \rightarrow u$ that for each $u \in U$ gives a subset $P \subset T$ that describes the interests of the users.

We can now define the two main parts of the problem, hence our goals.

Definition 3.3 (Topic extraction)**.** Given a set $D = \{d_1, d_2, ..., d_n\}$ where n is the number of tweets and d_i is the text of the i^{th} tweet, topic extraction consists of obtaining a value t_i that identifies the text d_i.

Definition 3.4 (User profiling)**.** Given a set $S = \{(u_1, t_1), (u_2, t_2), ..., (u_n, t_n)\}$, where n is the number of rows in a dataset, u_i is the i^{th} user and t_i is the t^{th} tweet, and interest sets $I_1, I_2, ..., I_m$, where the index $i = 0, ..., m$ corresponds to a single user, user profiling consists of obtaining data $d \in I_i$ foreach i, from 0 to m, where d has been retrieved by S.

Moreover, we decided to take advantage of the results of the two goals defined above in order to create a *recommendation system*, described below.

Definition 3.5 (Recommendation system)**.** Given a set $S = \{(u_1; t_1, ..., t_{k_1}), (u_2; t_1, ..., t_{k_2}), ..., (u_n; t_1, ..., t_{k_n}))\}$ where n is the number of users, u_i is the i^{th} user, and $t_1, ..., t_{k_i}$ are the topics of the i^{th} user, a recommendation system consists of generating an association $a \in R$ where R, the set of recommendation rules, is defined as

$$R = \{T_1 \rightarrow S_1, ..., T_m \rightarrow S_m\} \quad (1)$$

where each T and S are sets of topics, and in the presence of the set S_l the set T_l can be recommended.

4. SOLUTION

We began by analysing the dataset and its structure and defining which of the given data was useful for our solution.

The next phase was the cleaning of the data. Since the dataset we have is a collection of text generated by users in the context of social network, it is to be expected that it contains mostly noise due to typos and the use of abbreviations and slang. Generally, this type of information is not well interpreted by the already existing algorithms and, instead, is often considered erroneously as a possible topic.

After preprocessing the dataset we proceed to apply the *latent Dirichlet allocation* (LDA) in order to create a model of two topics represented each tweet. To the LDA model results were added the hashtags obtained from the parsing of the text of the tweets.

The fourth steps was to arrange the data in order to apply the FP-growth algorithm to retrieve the frequent itemsets

and following apply a mining approach to obtain association rules between topics. The results obtained in this step were defined as Recommendation-topic system.

The other step was the application of the k-means algorithm to obtain clusters of topics, and following cluster of users. Before applying the k-means algorithm, text data were arranged in the TF-IDF matrix in order to make possible the execution of the algorithm.

The results obtained from the fourth and fifth steps were analyzed in order to have a user profile. Clustering method give us a suggestion of users that have similiar interests and therefore that may become "friends" on the social media. Moreover, thanks to the results obtained from the FP-growth algorithm, is possible to identify those users that suggest topic to other users through the association rules.

4.1 Dataset Analysis

The given dataset is divided in different parts, a full dataset and a partial dataset. The former is further separated into a 01 and a 02 part. Each of these contains the following files:

- `user`
  This file contains information about Twitter users. In particular, it contains the ID and nickname of the user, information about privacy, namely whether the profile is protected or public; it contains also the date of creation of the account, the number of follower, friends, posted tweets of a user and the number of tweets he or she liked, along with information about the location, either given by the users themselves or automatically retrieved, like the offset from the UTC or the timezone. The users can also choose their preferred language and whether they want their tweets to be geo-tagged or not, insert a small bio about themselves and a link to a personal website, like an account in another social network or a portfolio. Finally, it contains a boolean that discriminated verified accounts. This field is useful to distinguish the real accounts of famous people from fan accounts or impostors.

- `tweet`
  This file contains metadata about tweets. Specifically, each tweet can be described by its ID and the ID of its author and the time of its creation. Its popularity can be inferred from the number of replies it received, whether it has been favourited or retweeted, and in the second case how many times. Moreover, the tweets are tagged with a language identifier that was machine-detected when possible.

- `tweet_hashtag`: this file contains the metadata about hashtags. It contains the fields:
  - `tweet_id`: ID of the tweet;
  - `user_id`: ID of the author of the tweet;
  - `hashtag_id`: ID or IDs of the hashtags in the tweet;

- `tweet_text`: it contains some metadata about tweets and their content. In particular, it is composed by the following columns:
  - `tweet_id`: ID of the tweet;
  - `user_id`: ID of the author of the tweet;
  - `text`: content of the tweet;

  Moreover, if the tweet is geotagged the following columns are not set to null and describe:
  - `geo_lat`: the latitude of the location from where the tweet was posted;
  - `geo_long`: the longitude of the location from where the tweet was posted;
  - `place_full_name`: the human-readable full name of the location;
  - `place_id`: an ID representing the place;

In this work, the 02 dataset was chosen, due to its shorter length. Specifically, only the `tweet_text_02.csv` and `tweet_hashtag_02.csv` files were used. This choice was determined by the fact that the former is the only file containing the content of the tweets, while the latter helped in locating the hashtags.

For our purpose, we chose to ignore the information about the geo-tagging of tweets for two reasons. First of all, we focused only on the extraction of topics from tweets and then, on this base, the creation of users' profiles. Therefore, we deemed the geo-location as not essential. Moreover, this knowledge is known to be sparse: out of about 11'100'000 tweets contained in the dataset only around 260'000 are geo-tagged. Similar results were also found by [5].

The only improvement that could be brought about by the `user` file could be that of visualising the users' profiles with their nicknames instead of their IDs. Therefore, we opted not to use this file.

On the other hand, the file `tweet` could help in creating statistics about tweets and maybe highlight those topics that are more interesting than others by exploiting the information about the number of replies to a given tweet or the number or retweets or likes a given tweet has received.

4.2 Preprocessing

The main step of our work was the preprocessing of the dataset. As previously mentioned, tweets, as other user created content, contain mostly noise. This is due to the informal context and due to the limit in the number of characters per tweet imposed by Twitter. For this reason, most users tend to use abbreviation and acronyms to express themselves.

In addition, to the extensive use of slang and abbreviations, it is important to notice that tweets often contain *URLs* and mentioning of other users with the tag @. This type of information could be useful to extract the relationship between users. However, this is not our purpose and hence we have decided to parse this knowledge out of our dataset, otherwise our algorithm could have interpreted such words as topics of tweets. Another important step of the preprocessing is that of removing the so called *"stopwords"*. This words are those that are very frequent for a given language and that almost never are indicative of the topic of a sentence. Examples of such words for English are "the" or "we".

For the reasons explained above, we chose to preprocess the tweets contents as follows:

- lowercase the sentences;
- remove the URLs;
- remove the mentions;
- tokenize the sentence;
- remove the stopwords from the tokes;
- remove the remaining words of only one character that could not be considered as a reasonable topic;
- *lemmatization* of the tokens. With lemmatization, we refer to a procedure that aims at transforming each word to their base form. For example, the verb term "is" is lemmatized to "be, whereas a plural noun such as "cats" is lemmatized to "cat". We decided to also POS-tag the tokens in order to achieve, when possible, a better lemmatization.
- remove the words that are composed by only numbers, with the exception of those of length between 2 and 4 that could be dates or years of interest.

To achieve this, we used the Natural Language ToolKit (NLTK) [6] library. However, we noticed that some terms, such as "follow", "retweet" or similar, were too frequent and therefore we created our own list of stopwords to extend the NLTK one.

4.3 Latent Dirichlet Allocation

In order to identify the main topics of a tweet text, the *Latent Dirichlet Allocation* generative statistical model has been used. LDA is an unsupervised machine learning technique that identifies latent topics from a section of a big document. It is based on the concept of "bag of words", hence each document is represented as a probability distribution over latent topics, while each topic is represented as a probability distribution over a number of words.

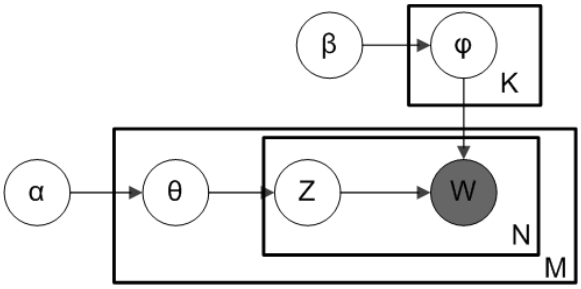

Figure 1: Graphical representation of the LDA model

As Figure 1 shows, each document M is associated to a multinomial distribution over topics, denoted as θ in the image, and each topic K is associated to a multinomial distribution over words, denoted as φ. Both have a Dirichlet prior with hyper-parameters α and β respectively. For each word in document M, a topic Z is extracted from θ and consequently a word W from φ is sampled. This process is repeated n times, where n is the number of words in the document.

4.4 Frequent itemsets and association rules

The frequent itemset approach is based on the "Market basket" problem. Starting from a list of customers and the items each of them has purchased, this problem aims at obtaining a list of frequent sets of items purchased by customers, therefore creating association rules between the frequent sets and the overall items.

There are many algorithms that allow to identify frequent itemsets in a list of items and they vary greatly in terms of computational cost and the accuracy of the results. The most known algorithms for this purpose are the Apriori algorithm, the Tree Projection, Relim and FP-Growth algorithm. For our work, the FP-growth algorithm was chosen for its better results and will be further explained in Section 4.4.1; whereas the association rules between the resulting itemsets have been created based on four concepts: support, confidence, lift and conviction. The details of the association rules mining are reported in Section 4.4.2.

4.4.1 Frequent itemsets

A frequent itemset can be defined as "a set of items that appears in many baskets". To be formal, we assume that there is a number s, called the *support threshold*. If I is a set of items, the support for I is the number of baskets for which I is a subset. We say I is frequent if its support is s or more. [7] This approach has been implemented using the FP-growth algorithm.[8] This algorithm is an efficient and scalable method for mining the complete set of frequent patterns using a tree structure called *FP-tree*.

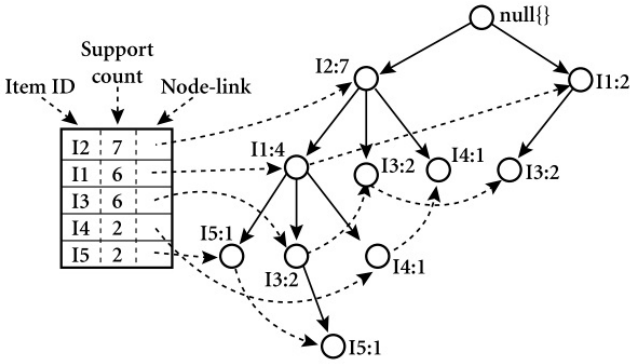

Figure 2: Example of FP-tree structure

Built before the application of the algorithm, the FP-tree stores quantitative information about frequent patterns in a database, which in our case is the set of tweets in the dataset. The FP-tree, as can be seen in the Figure 2, has one root labeled "null" connected with a set of subtrees and a frequent-item-header table. The subtrees represent the frequent itemsets and are composed by nodes interconnected by edges. Each entry of the frequent-item-header table consists of the *item_name* of the node, the *support_count* that is the number of occurrences of the *item_name* in the dataset, and

the head of *node_link*, namely a pointer to the first node in the FP-tree with that *item_name*.

The Algorithm 4.4.1 shows the FP-Growth Algorithm that takes as inputs the FP-Tree representing the text data and a *minimum support threshold* and returns a complete set of frequent patterns.

Algorithm 1 FP-Growth algorithm

```
1: procedure FP-GROWTH(FP-TREE, TRESHOLD
2:     if Tree contains a single prefix path then then
3:         P ← single prefix-path part of Tree
4:         Q ← multipath with top branching node = null root
5:         for each combinationbinP do
6:             s ← min support nodes ∈ b
7:             pattern b with support = s
8:         set(P) ← set of patterns generated
9:     else
10:        Q ← Tree
11:    for each i ∈ Q do
12:        s ← i.support
13:        pattern b with support = s
14:        build b's pattern-base
15:        build b's FP-Tree B
16:        if Tree B = ∅ then
17:            FP − growth(TreeB, b)
18:        set(Q) ← set of patterns generated
       return set(P) ∪ set(Q) ∪ (set(P) × set(Q))
```

In order to apply this algorithm to our data, it was necessary to create a matrix with a row for each user and a column for each of the topics extracted. Then the association rules were applied over the resulting frequent itemsets.

4.4.2 Association rules

The Association Mining rules problem has been defined by Agrawal, Imieliński, Swami. [7] Let us consider the "Market basket" problem. We can represent which items each customer has bought as a database of transactions T with each transaction t corresponding to a user. Each one of them is a binary vector that has a 1 in the position k if the customer t has bought item k.

Taking into consideration two itemsets X and Y, with Y that may be also equal to a single item I_l not part of X, we define an association rule A as

$$A : X \Rightarrow Y \tag{2}$$

In our example, such rules are satisfied if a customer t has bought the items represented in X and also the item or items represented by Y.

Moreover, it is necessary to explain the following concepts in order to better define an association rule between two sets of items.

Definition 4.1 (Support)**.** The term *support* indicates how frequently an itemset appears in the dataset. The support of X with respect of T is defined as

$$supp(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|} \tag{3}$$

Definition 4.2 (Confidence)**.** The term *confidence* is an indication of how often the rule is *true* in the application. The confidence is defined as

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \tag{4}$$

Definition 4.3 (Lift)**.** The term *lift* indicates the ratio between the observed support and the singular sets taken as independent sets. The confidence is defined as

$$lift(X \cup Y) = \frac{supp(X \cup Y)}{supp(X) \times supp(Y)} \tag{5}$$

Definition 4.4 (Conviction)**.** Assuming that X and Y are independent sets, the *conviction* of a rule is the ratio between the expected frequency that set X occurs without Y and the observed frequency of the incorrect prediction. It is defined as

$$conv(X \Rightarrow Y) = \frac{1 - supp(Y)}{1 - conf(X \Rightarrow Y)} \tag{6}$$

4.5 Clustering

To provide another analysis of the users of Twitter, a clustering approach has been implemented. The general idea was to cluster all the tweets that followed a common theme to obtain the list of the authors of these tweets. In order to achieve this, we first represented the tweets in a matrix over which was then applied the k-means algorithm.

4.5.1 TF-IDF matrix

The matrix mentioned above was created by applying the Term Frequency-Inverse Document Frequency (TF-IDF) scheme [9] over the preprocessed tweets. This scheme is divided into the following two steps:

1. **term frequency**: it generally consists in computing the number of appearances of a term in a document;

2. **inverse document frequency**: it is a measure of the specificity of the word, namely how rare a given word is in the collection of documents analyzed.

The measure obtained with the second step is used to down-weight the value term frequency of those words that appear too frequently in all the documents. With this method, it is possible to create a matrix as the one shown in Figure 3 to represent all the words present in the dataset as a vector.

Once the matrix has been built, the clustering algorithm may be applied.

4.5.2 k-means algorithm

The clustering algorithm used to group tweets, and consequently users, is the k-means algorithm. The k-means algorithm is "a method of vector quantization" [1]. The aim of the algorithm is to partition a number of observations, corresponding in our case documents, into k clusters in which "each observation belongs to the cluster with the nearest mean". Given a set of k initial centroids, the classical approach to the algorithm iterates over two steps:

1. **Assignment step**: assign each observation to the cluster whose mean has the least squared Euclidean

[1] https://en.wikipedia.org/wiki/K-means_clustering

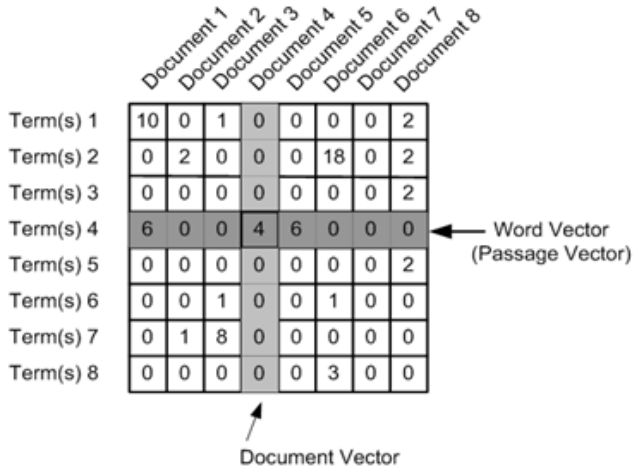

Figure 3: Example of a TF-IDF matrix

distance from a given centroid. Formally, given a collection c of centroids of cluster C, each data point x is assigned to a cluster based on

$$\arg\min_{c_i \in C} dist(c_i, x)^2 \tag{7}$$

where *dist(.)* is the standard Euclidean distance. The set of datapoint assignment for each i^{th} cluster centroid is S_i

2. **Centroid update step**: centroids are recomputed taking the mean of all data points assigned to centroid cluster. Formally

$$c_i = \frac{1}{S_i} \sum_{x_i \in S_i} x_i \tag{8}$$

The application of k-means on the TF-IDF matrix gives as result a list of clusters, defined by one or more centroids.

5. EXPERIMENTS

In this section, we will expose the results we obtained applying to our dataset the techniques presented in the Section 4.

5.1 LDA

Since the LDA model has been devised for the analysis of large documents, the performances and the results obtained with our data are bad when compared to other methodologies we have applied. As a matter of fact, each document of our collection is the content of a single tweet which has a very low number of words, as it is possible to see in Figure 4, and consequently the unsupervised model does not iterate the appropriate number of times, hence not giving quite good results.

The LDA model executed returned 2 topics, each composed by a single word. The number of topics to be retrieved has been hard-coded due to the length of each tweet. A choice of more topics per tweet would result to be redundant, whereas opting for the extraction of a single word does not allow to identify correctly the general topic of the tweet.

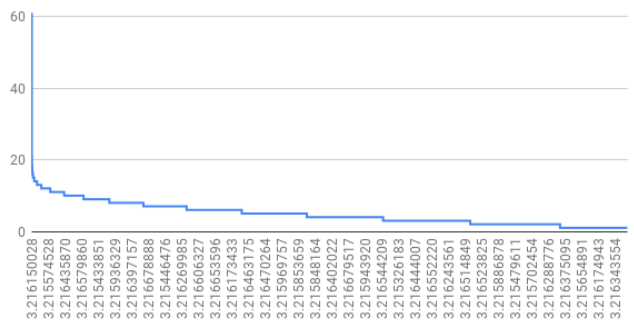

Figure 4: The number of words per tweet

Indeed, the word *good* has not a semantic meaning, but associated with the word *day* allows to better understand what the tweet is about.

In addition to the result obtained from LDA, the hashtags were also used to broaden the list of topics per tweet. The hashtag retrieval has been done parsing the text of the tweet and extracting the words tagged with the typical # of Twitter. The parsing of the text was necessary since in the dataset there is only the *ID* of the hashtag, but non the text.

5.2 Frequent Itemsets and Association Rules

Applying the frequent itemsets, we have obtained a list of itemsets like the one in the first column of Table 1. After applying the association rules we found, we obtained a list of expanded frequent itemsets like the one in the second column.

Frequent itemset	Associated Frequent itemset
collect, android	coin, androidgames, gameinsight
food, iphone	iphonegames, gameinsight
sagittarius	productive
impulse	challenge
concert	directioner
coin	collect

Table 1: Frequent itemsets

Once found the associated frequent itemsets, we can do inverse indexing in order to extract a list of topic tweeted by each user. By exploiting the frequent itemsets list we can suggest to the users new topics of interest. If users that talk about a certain topic, *topic_1* contained in a frequent itemset, they could be interested in a given *topic_2*, where the latter is the result of an association rule regarding the frequent itemset in question. Nevertheless, if the topics in the recommended itemset already are topics used by the user, the recommended list will be empty. In this way, we obtained a *recommendation system* based on the frequent topics twitted by different users, as it is possible to see in Table 3.

Moreover, it is possible to identify those users that could influence other users, simply associating to each frequent itemset those users that tweet about such topics in their

Frequent itemset	Users	Number of users
collect, android	maor_sherer, dark_lucifer_, Day_Oday	54
food, iphone	rimawevskaja, remdipsy, Meiyuei, ...	22
sagittarius	DosPabl_os, BALDIE_SAYS, lisaboag, ...	31
impulse	emull6789, Xhe_Is_Xanadu, , mzkenia, ...	41
concert	hannahgoffrock1, theneilcutter, _mellowlifee, ...	126
coin	Day_Oday, FrogsrKewl, JerryJerryrh71, ...	64

Table 2: Example of frequent itemsets and the number of users that posted them. Only some of the users were reported due to their number.

Users	Frequent itemset	Recommended topics
askemax	visit, winner, collect	coin
SK1NNYSHIT	bring, collect	coin
plfb007	coin, androidgames, collect, android	[ ]
boobear_98ele	mixer, directioner	concert
kartisn1644	concert, directioner	[ ]
karrouxche	buying, impulse	challenge
DosPabl_os	sagittarius, romance	productive

Table 3: Example of recommended topics for a group of users

tweets. This result, shown in Table 2, give us a first analysis of the users.

5.3 Clustering with k-means

In our application of the k-means algorithm over the TF-IDF matrix, we chose to extract 10 clusters whose centroid were initialised with random values. We fixed to 1000 the number of iterations and we obtained 10 final cluster whose centroids are shown in Table 4.

Cluster ID	Word 1	Word 2	Word 3
0	love	good	people
1	happy	day	new
2	bad	today	school
3	birthday	happy	people
4	think	life	girl
5	night	watch	need
6	twitter	tell	please
7	give	time	love
8	please	need	time
9	good	come	day

Table 4: Centroids found for the clusters

The results so obtained are quite unexpected but easily justifiable. As it is possible to notice, the centroids are generic words, like *love*, *new*, *good*, that may have different semantic meanings that, however, were not taken into account in our application. For this reason, the greatest number of topics found through the LDA in combination with the hashtags belongs to the Cluster 0, as it can be seen in Figure 5. Moreover, many topics belonging to this cluster are the most tweeted ones, as it is shown in Figure 7, and this explains why this cluster is the most populated one.

The membership of users to a cluster depends on their tweets: if a user tweets topics belonging to cluster x, consequently he or she belongs to this cluster. Thus, users membership directly depends on their tweets. The result of this association is shown in Figure 6 and it can be seen that even in this case the Cluster 0 results to be the most populated.

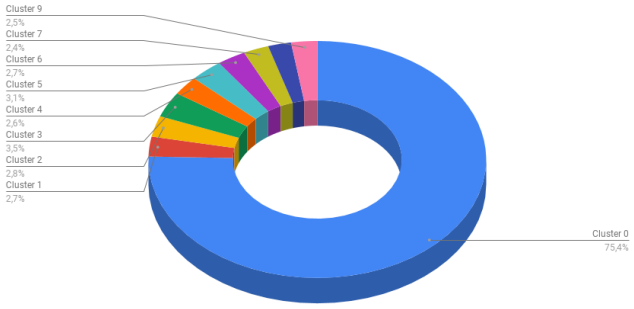

Figure 5: The number of topics per cluster

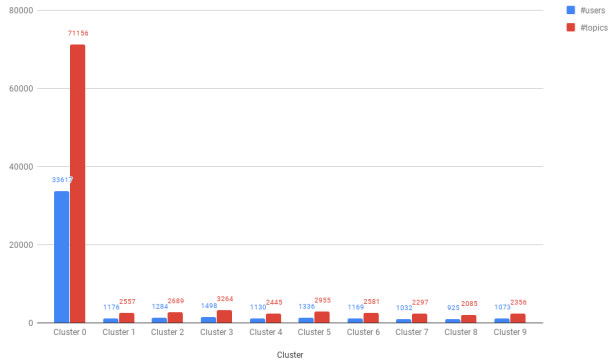

Figure 6: The relation of topics and user per cluster

6. CONCLUSIONS

Our work consists of the resolution of a problem composed by two main parts: to extract topics from a Twitter dataset and to find correlation among them; secondly to build a new user profile based on information extracted. For the first purpose, the solution we proposed consists of extracting the topics from a tweet and finding association rules among them to create a recommendation system. On the other hand, to create the user profile, we arrange the retrieved information

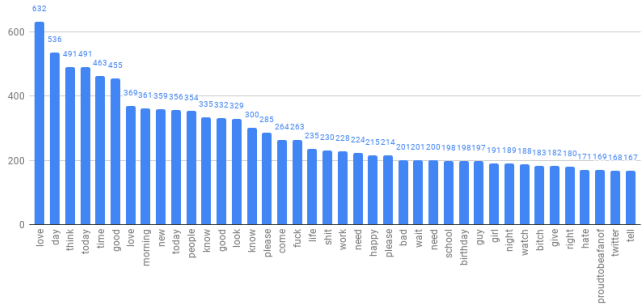

Figure 7: The number of tweets per topic

by frequent itemsets and clustering algorithms. The pipeline we proposed met some problems during the implementation. In particular, most of the problems derived from the structure of the text of the tweet that is noisy and short.

In the preprocessing phase, additional words were added to the *stopwords* dictionary to clean the text from specific terms used in the social media context, such as the words *rt, follow, f4f, tbt.*

The other problem was the application of the LDA algorithm due to the short structure of the text. Indeed, LDA has been built to work with standard documents so the performances obtained by the algorithm, even if they were satisfying, could be better if, for example, some tweets were grouped or re-writing an adapted LDA algorithm that works on short texts.

After the execution of the LDA, the frequent itemsets and the association rules were created and the algorithms performed as expected, giving the results shown in the previous section. The k-means algorithm, that allows us to build the user profile and reinforce the recommendation system gives quite good results because, as mentioned before, the texts are shorts and very different one from the others in term of contents and structure. To have better results, an approach similar to the one proposed to the "LDA problem" may be actuated. Even if we met some problems, the overall results are satisfying since we obtained a recommendation system of topics and a user profiling system based on the recommended topics and the clustering results, which were our prefixed goals.

7. REFERENCES

[1] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270. ACM, 2010.

[2] Xiaojun Quan, Chunyu Kit, Yong Ge, and Sinno Jialin Pan. Short and sparse text topic modeling via self-aggregation. In *IJCAI*, pages 2270–2276, 2015.

[3] Xin Wang, Ying Wang, Wanli Zuo, and Guoyong Cai. Exploring social context for topic identification in short and noisy texts. In *AAAI*, pages 1868–1874, 2015.

[4] Xiaohui Cui and Thomas E Potok. Document clustering analysis based on hybrid pso+ k-means algorithm. *Journal of Computer Sciences (special issue)*, 27:33, 2005.

[5] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 759–768. ACM, 2010.

[6] Steven Bird and Edward Loper. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics, 2004.

[7] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.

[8] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *ACM sigmod record*, volume 29, pages 1–12. ACM, 2000.

[9] Gerard Salton and Michael J McGill. Introduction to modern information retrieval. 1986.