Project 2017: Italian Referendum V2. (groups of 2 ± 1)

0) Temporal Analysis

1) Manually/Semi-Automatically collect on the web (political party website, institutional website, wikipedia, public available list of politicians) and collect all possible italian politics (or journalists) name/twitter account P; divide them in two group according to their support to Yes Y or No N (skip otherwise); How many users you get? How many tweets? Which is their distribution over time? (Provided list for groups of 1).

2) For each ***p*** in ***Y* | *N*** analyze all the tweets/retweet ***T(Y)*** and build SAX string (grain = 12h) for the Top 1000 words by frequencies that expose the typical pattern that capture the collective attention (see slides *SAX**); group together **(implementing a trivial *K-Means*)** all the strings that expose an equal temporal behaviour (same or very similar SAX string) ***t_1(Y), t_2(Y), t_1(N) , t_2(N)...***; (Clusters of terms)

3) For each group of token in ***t_i(Y)*** and in ***t_j(N)*** build the co-occurrence graph of them (two word ***t_1 , t_2*** have an edge ***e***, if they appear in the same document; the weight ***w*** of e is equal to the number of documents where both token appear); using a threshold over edge weights (decide which one produce best results) , identifying the Connected Components ***CC*** and extract the innermost core (*K-Core*) from each of them, producing subgroup of tokens, ***t_1'(Y),t_2''(Y)***.. Comment about the differences and decide which is the best strategy *K-Core* vs Simple *Connected Component*.
4) Using the original statistics (collection), trace the time series (grain 3h) for each obtained group of token ***t_i'(Y)***, ***t_j'(N)***; compare (manually) the time series of each group ***Y*** and ***N*** and comments about some possible kind of action-reaction that should be clearly identified. (look also at the content of the tweets) **(SKIP IT FOR GROUPS OF 1)**

1) Identify mentions of candidates or YES/NO supporter

1. From the entire tweets dataset, identify tweets of users that mention one of the politicians (include also the previously founded ***P*** account) that support YES or NO or directly express their opinion about the referendum (use also ***t_i'(Y)***, ***t_j'(N)*** groups of words). How many users you get? How many tweets? Let ***M*** be the set of such users and let ***T(M)*** be the set of related tweets.

2. Using the provided Graph and the library G (see slides to obtain it) first select the subgraph induced by users ***S(M)*** then find the largest connected component ***CC*** and compute HITS on this subgraph. Then, find the 1000 highest ranked (Authority) users. Who are they? Can be divided in YES and NO supporters? Propose some metrics.

3. Partitioning the users of ***M*** according to the candidates they mention (each user can mention more that one candidate more than one time). Identify the users mentioning more frequently each candidate or support YES/NO and measure their centrality (Hubness Authority). Find the 500 (for each option YES/NO) who both support the candidate frequently and are highly central (define some combined measure to select such candidates and propose a method to give sentiment to those mentions). Let Influencer ***M'*** in ***M*** be these users.

4. Identify for each option YES / NO which are the top 500 k-Players ***K*** using the KPP-NEG algorithm. (if it is necessary reduce the original graph removing those nodes with a degree lower than a selected threshold (48h of processing or so)). **(SKIP IT FOR GROUPS OF 1)**

2) **Spread of Influence**

1. Using a modified version of LPA (Label Propagation Algorithm start from the provided one in the G library) that start assigning a label only for those users that are classified with YES or NO estimates over the whole network which party spread more over the network. How is the spread over the network if:
   – only the identified k-Players ***K*** are used as seeds of the modified LPA?
   (SKIP IT FOR GROUPS OF 1)
   –Using ***M***
   –Using only the ***M'***

3) Addendum for groups of 3

2. Considering the subgraph S(**M**):
3. Decide a threshold over the edges to obtain an "interesting" number of comunities using LPA (propose/use a metric to measure what "interesting" mean).
4. Runnning LPA severals times on S(**M**) find subcomunities.
5. Propose a function (over the LPA runs) to decide finally to which comunity a user u belongs to.
6. Considering the detected comunities compute the average symilarity (cosine) of each one consider only the friends that are inside the population **P**.
7. Commenting the top 10 comunties. How the comunities are polarized and how?

4) **Dataset**

The provided Dataset contain a sampled network composed by TwitterID
(one edge per line *src***<tab>***dst***<tab>***weight*):
Official_SBN-ITA-2016-Net.gz
and the folder ***stream*** that containing the stream across 4th of December divided by day;
each files contains 10000 tweets.

The dataset (around 10Gb) is Available using Resilio Sync Software:
https://www.resilio.com/individuals/
with the following key:
BGO6J6CQDRKBEDG3HRFXRN45TLXST3ERW

5) IMPORTANT NOTES:

You MUST use Java

here are some minimum requirements expected for the project:

- the project must be organized in Maven format with all the necessary dependencies (specified in the pom);
- the project must be successfully compilable;
- the project must contain relative path to the required data;
- the project must use the technologies exposed during the course;
- the project should save intermediate elaborated data;
- the project must contain 1 or more main method to complete the workflow;
- you must provide the code of every step (the project must be complete).

The project must be READY to be run without special knowledge or particular effort.
The student need to provide a report that describe each step and the obtained results;
which problem and which solution are provided to solve them.