Report of the Web&Social Information Extraction project: Mining Evolving Topics

Manuel Prandini ID: 1707827

I. INTRODUCTION

The project aims to identify and track topics for each year, from 2000 to 2018, within a network formed by keywords, one for each year. A *topic* can be seen as a set of keywords that are more prominent in a specific year's network. *To trace*, we intend to study and see how a topic changes shape over a period of time. For example, take an argument *t* in a year *y*, the aim is to see if in the years *y+1, y+2, .., y+n* the keywords of the topic change, remain the same, decrease or increase. The project was divided into several phases: the preprocessing of two datasets, the construction of a graph composed of keywords for each year (from 2000 to 2018), the identification of the most important keywords within each graph, the identification of the topics of each year through an Spread of Influence Algorithm, starting from the main keywords of the year in question and finally tracing topics in consecutive years.

II. PREPROCESSING

For the project, two datasets were used: one containing data relating to the co-occurrences of the keywords in the networks of the relative years, called *'ds-1.tsv'*, the other containing data relating to the collaborations carried out between the various authors in the relative years, called *'ds-2.tsv'*. Both datasets contained lines of information even for years that were not of interest to the finalization of the project, so they were removed. Furthermore, the file 'ds-1.tsv' also contained lines with a series of characters *'???'* instead of keywords, so those have also been removed. For greater convenience, datasets have been organized in smaller rows, where the rows have been grouped for each year, from 2000 to 2018.

III. GRAPH CONSTRUCTION

19 undirected weighted graphs were built, one for each year, starting from reading the 'ds1.tsv' file of the corresponding year. Each graph *G* [Fig:1] is therefore composed of the nodes that correspond to the keywords of that year, and of arcs indicating whether two keywords are co-occurring within an article of some author.

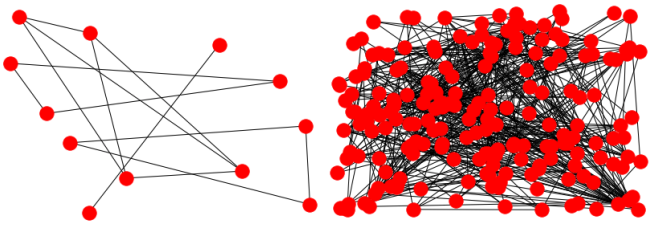

Fig. 1: The figure to the left represent the 2000's graph. The figure to the right represent the 2018's graph. It's possible to see the difference of keywords along this time interval.

As for the weights, they were set using a weighted sum defined as follows:

$$weight(k1, k2) = \sum_{a=1}^{A} TKU(a) * collaborations(a)$$

Where: *'TKU(a)'* is the number of times the author *'a'* uses the two keywords *k1, k2* within his article, and *'collaborations(a)'* is the number of collaborations performed by the author with other authors. I used for each year the total collaborations accumulated by an author until at that year because during the years an author can improves his collaborations with other authors. This last information is obtained from the 'ds2.tsv' type dataset. In this way, a keyword is relevant within the network if, in addition to being in many co-occurrences with other keywords, it has been used by authors who are very willing to collaborate and therefore important. For each graph then, the weights have been normalized, scaling them between the interval [0.0,1.0] using the *MinMax Normalization* in this way:

$$y = \frac{(x - min)}{(max - min)}$$

where *min* and *max* are the minimum and maximum weight values, and *x* is the relative weight. The normalization was important in order to then set a threshold

between 0.0 and 1.0 for each node of the graph during the iteration of the *Spread of Influence algorithm*, otherwise with weights greater than 1 all the nodes would have been influenced. In addition I tried to make each graph inherit the nodes, arcs, and weights of the previous graphs, thus constructing a graph for each successive year that is always larger, but it did not bring great results because then, when I was going to calculate the topics for each year, the information of the past years was brought back, not allowing to focus for each year which keywords could be most influenced and could be part of those topics.

IV. TOP KEYWORDS IDENTIFICATION

By *k-top keywords* of a year we mean the k-keywords (where k is an integer greater than 0 and less than the cardinality of the nodes of the graph) within the relative year graph that get a higher score according to a given measure. To identify the k-top keywords, different metrics were tried, including *PageRank*, *Authority Score*, *Hubness Score*, *Degree Centrality* (using weights), *Closeness Centrality* and *Betweenness Centrality* to see which was the best result. Each metric has always reported different keywords, but since our graph is non-direct, it is more correct to use *Centrality-based* metrics conceptually. Since weights are important in our graphs, it has been opted for the final choice for the Degree Centrality also using weights. I tried to modify also the parameter *k* for the selection of the top keywords but I didn't take a *k* major than 10 because the 2000's graph has only 14 nodes.

V. T1: TOPIC IDENTIFICATION

We said earlier that a topic [Fig:3] of a given year is a set of very relevant keywords within the corresponding year graph. The topic is formed by giving the set of k-top keywords defined above to the *Spread of Influence Algorithm*. Each node that is not active and which has an active node as its incoming arc, is influenced by the weight of the edge and if the sum of the weights of the relative edges is greater than the threshold of the node, it is influenced, added to the set of nodes active and also added to the set of keywords forming the topic. Each topic, therefore, has a set of nodes influenced by the top keyword and in turn influenced other nodes. Tests have been carried out using the algorithm starting from a keyword at a time, but this has produced topics that are too similar and very large, resulting inefficient. The best results (topic more reasonable) came out running the algorithm once and giving the whole set of k-top keywords as input. At the end of each year, k topics were produced. I set a randomic number between [0.0,1.0] to the threshold of the nodes, and i supposed that each node has a certain probability to be influenced. This because setting different thresholds, the result of topics are always different and become impossible to say which of these is correct.

```
label: digital images
set:
 {'digital images', 'increasing demand', 'images available'}

label: efficient method|rapid growth|digital images
set:
 {'digital images', 'increasing demand', 'images available', 'rapid growth', 'efficient method'}
```

Fig. 2: In the figure above the topic generated for a year is shown. It is composed of the top-keyword as a label, and a set of keywords that are the nodes affected body of the topic. The figure below shows the topic merged with other similar topics of the same year, where there are concatenated top-keywords as label.

A. Hierarchical Agglomerative Cluster Algorithm

Once the topics of a specific year were produced, I tried to merge [Fig:3] them by trying three metrics: *Jaccard similarity*, *Cosine similarity* and *Overlapping similarity*. After several attempts, I noticed that the Overlapping similarity for some topics behaved better. The formula is:

$$Overlapping(A, B) = \frac{A \cap B}{min(|A|, |B|)}$$

Also for the similarity threshold, I did several tests and in the end, I set the threshold to the 0.55 as it produced better more reasonable merged topics. I adopted this metric within the Hierarchical Clustering Algorithm and went on to create clusters (or merged topics) until a couple was similarly more than a certain threshold. So at each iteration, I compared all the possible pairs of topics looking for the most similar pair and that it was even higher than the minimum similarity threshold, once I found I merged the two topics and I concatenated the keys (top keywords), and I repeated the process with the new merged couple and the remaining topics. The algorithm ends if in an iteration no merge could be found to be performed.

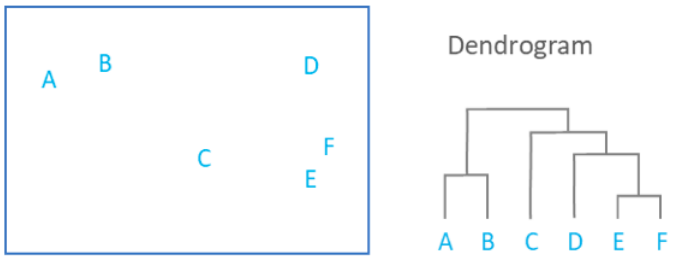

Fig. 3: Hierarchical Agglomerative Clustering represented as dendrogram. The letters correspond to the topics, and in each iteration, we search for the distance (in this case the Overlapping similarity) between all the pairs of topics. The couple with the highest score is merged into a single cluster (larger topic) and the process is repeated until bigger topics are created..

VI. T2: TOPIC TRACING

The final objective of the project was that, once the topics were found for each year from 2000 to 2018, they had to be merged based on their inter-annual similarity, and had to be included in a final list. Also in this part, I tried for each topic of a year y, to find the best match with a topic of a consecutive year y + 1. I moved in two directions: in the first, for each pair of topics in two consecutive years, I chose the best coupling based always on the Overlapping similarity of the keyword sets and taking only the pairs that had a score greater than zero. In the second, for each pair of topics in two consecutive years, I calculated the best Overlapping similarity on the pairs of edges of the two induced subgraphs, created only with the common nodes of the two topics, again taking only values greater than zero. These two approaches have produced two different results (they are shown in the 'results' folder of the project). Certainly, the second approach is a more accurate way of controlling similarity in that when speaking of graphs, we study the isomorphism between them. A better approach would be to calculate the *Edit distance*, but trying the algorithm on the biggest graphs, I noticed that it does not converge. This algorithm try to transform one subgraph to the other by doing a number of operations (additions, deletions, substitutions of nodes or edges, and reversions of edges). Each operation has a cost and this method try to find the sequence of operations that minimizes the computational cost of matching the two subgraphs.

VII. TOPIC ANALYSIS

After creating the various topics for each year from 2000 to 2018, it was possible to go and make some statistics.

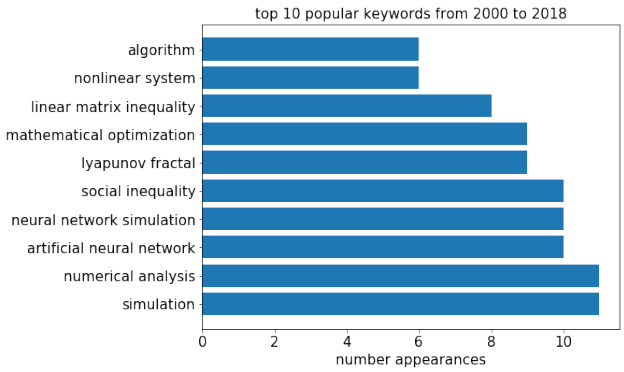

Fig. 4: Here are shown the top 10 keywords most popular that are present inside the topics from 2000 to 2018.

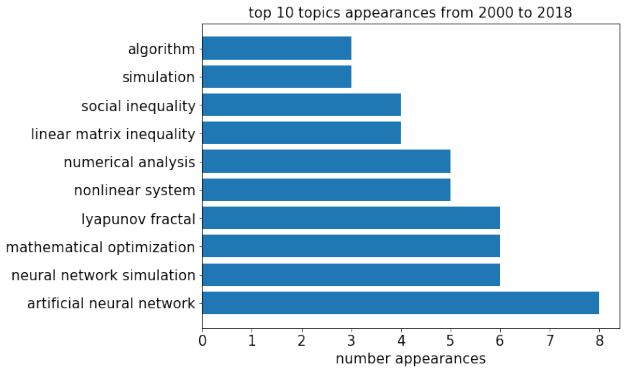

Fig. 5: Here are shown the top 10 topics (based on labels) that appear in more years from 2000 to 2018.

VIII. CONCLUSIONS

After all the operations carried out and said in the previous sections of the report, two results were produced (visible in the folder 'results'). After several tests, trying to combine the best techniques to generate reasonable topics, both choosing the weights to attribute to the graphs, and choosing the metric to select the top keywords, and modifying the Spread of Influence Algorithm (starting with a keyword at a time, starting with all the keywords together), and by merging the topics with different similarity metrics and with different thresholds. Logically, in the graphs with fewer nodes, topics formed even by the top keyword itself were produced, that is that they did not influence other

nodes. In the graphs with more nodes, more words were influenced and during the merge phase, large topics were produced, perhaps also due to the low threshold. For the threshold, however, a compromise had to be chosen that merged even the smallest topics. For the inter-annual merge, the best pairs resulted in the largest graphs. If different topics are produced each time the previously mentioned metrics are changed, it is due to the fact that the two datasets provided have little data, and moreover the words are related to each other only if they have a co-occurrence to the internal of an article by an author, but data on articles have not been provided, and it would also be advisable to make sense of the keywords through a *Word Embedding* process that attributes words to a vector space and tries to bring vectors whose words they have a more similar semantic meaning.