Cluster approach to analysis of publication titles

Arina Bogomolova¹, ², Maria Ryazanova² and Igor Balk²

¹ Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russia
² Global innovation Labs LLC, 258 Harvard Str #352 Brookline MA 02446 USA

e-mail: info@innovationlabs.net

Abstract. Text analysis is a promising field of study with many unsolved problems. First of all, most methods are labor and time consuming. We want to pay special attention to patents. The most important thing in analyzing patents as a reflection of a company's research activities is not to be late. Technology is emerging very quickly. So speed of response to changes in the world of scientific research is very important now. Therefore, we propose an alternative method of patent analysis based on clustering. Its main advantage is that it does not require different train/test datasets and it could be applied immediately. In this article, we compare different clustering algorithms, because the quality of the conclusions depends on it.

1. Introduction

The issue of textual information analysis is a complex and urgent task. Text has more complex structure than numeric or categorical data, thus it is more difficult to analyze by machine learning methods. Nevertheless, in the age of technology dominance, it is necessary to use all available information, including texts, which contains a great wealth of knowledge.

For example, the analysis of patents is a very important task for planning the future of a company and its research activity. Thousands of new patents are publishing every day and shape the future technological development agenda. So an analysis of patents and publications is an opportunity to find technologies that would be relevant in a 2–5 years.

There are many methods of text analysis [1]. First of all, it is Word2Vec method, then goes Long-short term memory classification model, K-Nearest Neighbor classification model, Convolutional Neural Network classification model and other models based on these [2]. However, analyzing full texts is a rather difficult task that requires large computational resources and time-consuming data cleaning. For most methods, it is necessary to have a training sample. The task is also to compile an extensive dictionary of stop words. Semantic vocabulary can take years as well as model training could be not an easy task. Semantic network created from patent documents is a successful patent analysis method [3]. But full texts of patents still require significant effort in processing the keyword dictionary.

Contrariwise, we can use the titles of publications and patents to get the result quickly and efficiently. Although this data type also has its own characteristics. First, the titles briefly reflect the main topic of the research, but do not always give an accurate answer about the results. Second, there are some patterns in the titles composition, it is especially evident in the example of patents. Therefore, it is important to take into account that the words such as “method” or “apparatus” will not add any value for understanding of the topic. Thirdly, words with a high frequency are more interesting for us, than words that occurred only once in the entire sample. Most often, such unique tokens are excluded from
consideration. Consequently, the set of tokens under study will not get much bigger with sample size growth. Only the frequency will grow steadily.

The main idea is to cluster tokens from patent titles to see the context of the most popular words. Here we come to another equally important question which token clustering algorithms to use to see topics of current interest. There are many articles on comparing different algorithms [4]. Nevertheless, we would like to apply clustering algorithms on our dataset and analyze the results, taking into account the peculiarity of the data and our goals. We also want to test the new Leiden algorithm, which has not yet become popular [5].

2. Methodology and data
In this study we compare clustering algorithms by applying them to a dataset from open source – patent database [lens.org]. We selected Apple Inc. patents published between January 2019 and September 2020 as an example and found 13.5 thousand patents on various topics. The first step in patent analysis was data cleaning. For this purpose, we used a set of Stop-words, which includes conjunctions, prepositions, articles, etc. We also removed all punctuation marks and words that have no semantic meaning. The next step is tokenization and network building. Each word has become a separate token (node) linked to other tokens via edges. There is an edge between two nodes if they meet in the joint title. Moreover, each node has a feature such as frequency, which reflects the number of tokens in the study sample. And each edge between two nodes shows how many times these two tokens have been used together in the same title. Thus, we get a weighted undirected network of tokens.

The next step is to merge these tokens into clusters. The main idea is to reconstruct the topics of patents and highlight the general directions of research by summarizing token groups. Thus, the quality of clustering is significant for our task. Let’s compare several clustering algorithms, we described below.

2.1. Louvain algorithm
The Louvain algorithm is a well-known method of community detection in large-scale networks (consisting of millions of nodes and edges). It was first introduced in 2008 by Blondel et al. [6]

Let’s suppose we have a network. In this case we usually want to gather some information about it and a good way to do it is by finding a way to split the network into modules (communities), with each module’s nodes being strongly connected, and different module’s nodes being sparsely connected. The strength of such division is measured by a quality function. It should be maximized in order to obtain a better partition. In general quality function is written as shown below

$$H = \sum_{ij} \left( a_{ij}A_{ij} - b_{ij}(1 - A_{ij}) \right) \delta(\sigma_i, \sigma_j)$$  \hspace{1cm} (1)

where

- $A_{ij}$ is the adjacency matrix element: $A_{ij} = 1$ if there is an edge between nodes $i$ and $j$, and 0 otherwise
- $\sigma_i$ is the community which node $i$ belongs to
- $\delta(u, v)$ is the Dirac delta function ($\delta(u, v) = 1$ if $u = v$, $\delta(u, v) = 0$ if $u \neq v$)
- $a_{ij}, b_{ij} \geq 0$ are weights-coefficients.

Most commonly used quality function is called modularity of the partition – a scalar value which lies in the range $[-1, 1]$. It measures the density of edges between nodes inside modules as compared to the density of edges between modules. By making $a_{ij} = 0$ and $b_{ij} = (k_i k_j)/(2m)$ in (1) we get modularity. For weighted undirected networks modularity of the partition is defined as [7].

$$M = \frac{1}{2m} \sum_{ij} \left( A_{ij}\omega_{ij} - \frac{k_i k_j}{2m} \right) \delta(\sigma_i, \sigma_j)$$  \hspace{1cm} (2)

where
\[ A_{ij}, \sigma, \delta(u, v) \text{ as in (1)} \]
\[ \omega_{ij} \text{ is the weight of (ij) edge} \]
\[ k_i = \sum_j A_{ij} \omega_{ij} \text{ – the weighted degree of node } i \]
\[ m = 1/2 \sum_j A_{ij} \omega_{ij} \text{ – the sum of the weights of the edges in the network.} \]

Straightforward modularity maximization is computationally hard. Louvain algorithm for detecting communities in networks (including large-scale networks) offers a decent solution to this problem. It works by taking multiple passes at the dataset, each pass consisting of two steps: modularity optimization and community aggregation [6].

First step is to
a) create singleton partition by assigning each node to its own community
b) calculate modularity gain for a node’s possible replacement into one of the communities its immediate neighbors belong to
c) reassign the node to the community which provides maximum modularity gain
d) repeat (b)-(c) for all nodes sequentially until no further increase in modularity can be attained.

The gain in modularity \( \Delta M \) (can be calculated as \( 2 \Delta M m \)) after placing node \( i \) into a community \( \sigma \) is shown below [6]. It is easy to compute, so the algorithm is fast. A similar expression is used to calculate \( \Delta M \) of removing \( i \) from its community.

\[
\Delta M = \left[ \frac{\Sigma_{in} + k_{in}}{2m} - \frac{\left(\Sigma_{tot} + k_i\right)^2}{2m} \right] - \left[ \frac{\Sigma_{in}}{2m} - \frac{\left(\Sigma_{tot} - k_i\right)^2}{2m} \right]
\]
\[
\Delta M = \frac{k_{in} - 2 \cdot \Sigma_{tot} \cdot k_i}{(2m)^2} \rightarrow 2 \Delta M m = k_{in} - \frac{\Sigma_{tot} \cdot k_i}{m}
\]

where
\[ \Sigma_{in} \text{ is the sum of the weights of the edges in } \sigma \]
\[ \Sigma_{tot} \text{ is the sum of the weights of the edges incident to nodes in } \sigma \]
\[ k_i \text{ and } m \text{ as in (1)} \]
\[ k_{in} \text{ is the sum of the weights of the edges between } i \text{ and nodes in } \sigma \]

The second step of the algorithm is to take the communities found in the first step and address them as nodes of a new network. The weights of edges between nodes of the new network are calculated by summing up the weights of all initial edges between nodes of the corresponding communities.

The goal is achieved when each additional pass no longer raises the modularity. It is a so-called greedy optimization method with computational time \( O(n \log n) \), where \( n \) represents the number of nodes [8].

The Louvain algorithm is fairly fast and yields good results. However, frequently the communities it detects may be badly connected or even disconnected [9]. It also suffers from the resolution limit [10].

Apart from modularity there is another quality function that can be maximized for quality community detection called CPM (Constant Potts Model). By making \( a_{ij} = \omega_{ij} - b_{ij} \) and \( b_{ij} = \gamma \) in (1), where \( \gamma > 0 \) is the resolution parameter, we get the Constant Potts Model (shown below) [11]. Resolution parameter functions as a sort of threshold: communities should have a density of at least \( \gamma \), while the density between communities should be lower than \( \gamma \). Higher \( \gamma \) leads to more communities and vice versa.

\[
H_{CPM} = \sum_{ij} (A_{ij} \omega_{ij} - \gamma) \delta(\sigma_i, \sigma_j)
\]

It can be rewritten as

\[
H_{CPM} = \sum_c (e_c - \gamma n_c^2)
\]
Where $e_c = \Sigma A_{ij} \omega_{ij} \delta(\sigma_i, c) \delta(\sigma_j, c)$ – the total weight of edges inside a community $c$, $n_c = \Sigma \delta(\sigma_i, c)$ – the total number of nodes in $c$.

CPM maximization helps solve the problem of the resolution limit, but does not improve the Louvain algorithm with regard to badly-connected (or disconnected) communities detection.

2.2. Leiden algorithm

The Leiden algorithm, first introduced in 2019 by Traag et al., builds upon the Louvain algorithm. It is significantly faster and finds high-quality well-connected communities thus solving the problem of badly connected communities present in the Louvain algorithm [9].

The Louvain algorithm focuses on merging only, even when it leads to formation of badly connected communities where, for instance, only one node can be the link between two parts of the community. Unlike its predecessor, the Leiden algorithm can split clusters ensuring that all of them are well-connected.

The Leiden algorithm consists of three steps [9]:

1) singleton partition and local movement – similar to the Louvain method, but instead of checking all nodes one by one for possible community replacement all nodes which surroundings have stayed the same are ignored

2) refinement of the partition – finding subcommunities by regarding each initial community as a network and implementing the first step in it (the difference here is that moving nodes to communities does not necessarily maximize the quality function $H$ (modularity or CPM), but rather increases it – the probability of such replacement is higher when $\Delta H$ is higher – it allows for better partition)

3) aggregation of the network: non-refined partition is what the Louvain algorithm would then consider as a node of a new network, whereas the Leiden algorithm considers it as a community where subcommunities found after refined partition become nodes.

At no step is the quality function allowed to decrease. After several passes consisting of these three steps the clusterization can be improved no further (communities are subset optimal) and the process stops.

2.3. Label propagation algorithm

The label propagation algorithm (LPA) was introduced by Xiaojin Zhu and Zoubin Ghahramani in 2002 [12]. The version of LPA we review here belongs to Raghavan et al. [13]. Nodes inside a community are strongly connected while different communities are weakly connected. The name of the method speaks for itself – at first each node is assigned a unique name – label, then labels get propagated through the network: at each pass nodes in a random order one by one change their labels to the labels most common in their neighbourhood. If there is a tie it gets broken randomly. The process stops either when it has gone through a set maximum number of iterations or when each node has got the label prevailing in its neighbourhood. At this point most labels have already disappeared and the network is divided into communities.

LPA works locally, its computational time on a network with $e$ edges is linear – $O(e)$, so it is very fast [9, 10]. LPA is also easy to implement, but it has disadvantages. It produces multiple solutions for the same starting parameters [15]. It may also produce an extremely imbalanced partition with a giant community covering almost the whole network [16].

LPA is a semi-supervised algorithm. It does not need any prior information about the communities to work. However it is possible to assign labels to some nodes manually. The purpose of this is to lower the number of possible solutions provided by LPA.

Although LPA does not operate based on some quality function maximization it should be noted that we arrive at LPA by setting $a_{ij} = \omega_{ij}$ and $b_{ij} = 0$ in $H$ (or setting $\gamma = 0$ in $H_{CPM}$) [15].

$$H_{LP} = \sum_{ij} A_{ij} \omega_{ij} \delta(\sigma_i, \sigma_j)$$
Mathematical notations are the same as in (1) and (2).
The global maximum is a trivial solution where all nodes belong to the same community. Local maxima are the ones found with LPA.

2.4. Girvan-Newman algorithm
The Girvan-Newman algorithm was first introduced in 2002 by Girvan and Newman [17]. Similar to the Louvain algorithm it is a hierarchical method for community detection.

The algorithm works according to the following scheme:
1) calculate the edge betweenness coefficients for all edges of the graph (edge betweenness is defined as the number of shortest paths between all pairs of nodes passing along a given edge)
2) remove the edges with highest coefficients – remaining groups of connected nodes are considered communities
3) recalculate edge betweenness coefficients for all edges affected by removal (2)
4) repeat (2)-(3) until there are no edges left.

As a result a hierarchical structure of the network is obtained in a form of a dendrogram. To get a look at the communities in the network we can measure the modularity of the partition at each step and choose the step at which the modularity is maximized.

This algorithm has many modifications, which boil down to calculating other edge coefficients or replacing modularity with another similar quality function. Its main drawback is the running time, since calculating the edge betweenness coefficients for all edges is a computationally difficult task. Computational complexity of the Girvan-Newman algorithm is $O(m^2n)$ [14]. Using this algorithm on large-scale networks is not recommended.

3. Comparison of clustering algorithms
So, let's move on to the next step of our research and apply these clustering algorithms on our patent titles study sample. For these purposes we use Gephi, that is a software package for network analysis and visualization. To illustrate the results more clearly, we limited the number of tokens by frequency. Thus, in the pictures below, you will see a network of words that are used more than 40 times in the study data set. There are 296 tokens left on the network and they are connected by 8773 edges. We ended up with a very dense network, where almost all nodes are connected to each other. Such networks can also be found in other clustering tasks, as analyzing social groups in a small organization, for example.

We assume, there are several big research directions. Each research direction has its own set of keywords – tokens. Publications include the same keywords if they are connected to one big topic. The algorithm we are looking for should help us find these groups of words, that belong to the same topic.

We started with the Label propagation algorithm, that gave us 17 clusters. However, these clusters are not balanced. We have one large (red) group of tokens that unites the majority and the rest of the clusters consist of 1–10 tokens.

A similar result was given by using Girvan-Newman algorithm. We found 125 clusters, where one is majority, and the rest of the clusters contain 1 token each. Obviously, this clustering method does not suit our task. If we were looking for outsiders in social group, then perhaps these methods would be useful. But we want to get clusters of research topics. Therefore, we move on to the following methods.

Leiden and Louvain algorithms have a lot in common, so we will compare them using different quality functions. Let's start with Constant Potts Model (CPM). The new clustering result has one large main cluster and many smaller ones again. Both algorithms have similar cafeteria metrics Quality (0.42) and formed 37–39 clusters. This result is already better than we’ve got after the Label propagation and Girvan-Newman algorithms, because there are more clusters of medium size that can characterize a particular research topic. Nevertheless, we still cannot draw conclusions on research topics that we are interested in.

But the use of modularity as a quality function gave its results. Finally, the network is divided into more or less equivalent groups of tokens, the combination of which gives meaningful interpretations. Before drawing conclusions about Apple's developments, it should be noted that both algorithms gave a
resembling results. They have the same quality (0.32), and the clusters are identical in constitution with the exception of a few tokens and one additional cluster on Louvain side. Based on the work of the algorithms, we expected a better result from new Leiden algorithm, however, we cannot draw an unambiguous conclusion in our case due to highly similar composition of clusters.

**Figure 1.** Label propagation algorithm  
Number of clusters: 17

**Figure 2.** Girvan-Newman algorithm  
Number of clusters: 125

**Figure 3.** Leiden algorithm  
Quality Function: Constant Potts Model (CPM)  
Quality: 0.424  
Number of clusters: 37

**Figure 4.** Louvain algorithm  
Quality Function: Constant Potts Model (CPM)  
Quality: 0.423  
Number of clusters: 39
Analyzing the results obtained, we can assume that Apple has most of researches over the past year and a half in the following areas: communications, physical interface and electronics, user interface and virtual assistant.
4. Conclusions and future research

We have concluded that the most important element is the use of Modularity as a quality function for Louvain and Leiden algorithms. For our purposes, we cannot make an unambiguous conclusion which of these algorithms is better. Therefore, any of them could be used. Taking into account that networks with a large number of connections are encountered not only when analyzing texts, we can also draw conclusions about clustering algorithms. For example, to identify the dominant group, methods that use CPM as a quality function are more suitable. However, to analyze titles, Leiden or Louvain with modularity are better. An equally important part is the interpretation of the results, so the method is not devoid of expert element. But the main advantage of this method of analyzing patents is its speed, simplicity and universality. So this method is the best choice for quick assessing the situation in a certain company or field of activity (we used this method to study trends in renewable energy [18]). Further research will be related to improving data cleaning methods and simplifying the analysis of longer texts using the same clustering method. We are also interested in reducing the expert element in the decision algorithm, but without wasting advantage of low time consumption.

References

[1] Kim J M, Kim N K, Jung Y and Jun S 2019 Patent data analysis using functional count data model Soft computing 23 (18) pp 8815–26
[2] Xiao LZ, Wang GZ and Zuo Y 2018 Research on patent text classification based on Word2Vec and LSTM 11th international symposium on computational intelligence and design (iscid) 1 pp 71–74
[3] Kim Y G, Suh J H and Park S C 2008 Visualization of patent analysis for emerging technology Expert systems with applications 34(3) pp 1804–12
[4] Emmons S, Kobourov S G, Gallant M and Borner K 2016 Analysis of Network Clustering Algorithms and Cluster Quality Metrics at Scale PLoS ONE 11(7) DOI: 10.1371/journal.pone.0159161
[5] Mkhitaryan K, Mothe J and Haroutunian M 2019 Detecting communities from networks: comparison of algorithms on real and synthetic networks International Journal Information Theories and Applications 26 http://www.foibg.com/ijita/vol26/ijita26-03-p03.pdf
[6] Blondel Vincent D, Guillaume Jean-Loup, Lambiotte Renaud and Lefebvre Etienne 2008 Fast unfolding of communities in large networks Journal of Statistical Mechanics: Theory and Experiment 10 DOI: 10.1088/1742-5468/2008/10/P10008
[7] Newman M E J 2004 Analysis of weighted networks Phys. Rev. E 70 056131
[8] Sara Rahiminejad, Mano R. Maurya and Shankar Subramaniam 2019 Topological and functional comparison of community detection algorithms in biological networks BMC Bioinformatics 212(20) https://doi.org/10.1186/s12859-019-2746-0
[9] V A Traag, L Waltman and N J van Eck 2019 From Louvain to Leiden: guaranteeing well-connected communities (Scientific Reports 9, Article number: 5233)
[10] Fortunato S and Barthélemy M Resolution Limit in Community Detection Proc. Natl. Acad. Sci. USA 104 36
[11] V A Traag, P Van Dooren and Y Nesterov 2011 Narrow scope for resolution-limit-free community detection. Phys. Rev. E 84 016114
[12] Xiaojin Zhu and Zoubin Ghahramani 2002 Learning from labeled and unlabeled data with label propagation (School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Technical Report. CMU-CALD-02–107)
[13] Usha Nandini Raghavan, Réka Albert and Soundar Kumara 2007 Near linear time algorithm to detect community structures in large-scale networks https://journals.aps.org/pre/pdf/10.1103/PhysRevE.76.036106
[14] Ionkin M S and Ogneva M V 2017 Software implementation, efficiency analysis and quality assessment of algorithms for clustering graph models of social networks Izvestiya
Saratovskogo Universiteta Novaya Seriya – Matematika Mekhanika Informatika 17(4) pp 441–451

[15] G Tibely, J Kertesz 2008 On the equivalence of the label propagation method of community detection and a Potts model approach Physica A: Statistical Mechanics and its Applications 387(19–20) pp 4982–84

[16] Xuegang Hu, Wei He, Huizong Li and Jianhan Pan Role-based Label Propagation Algorithm for Community Detection

[17] Girvan M, Newman M E J 2002 Community structure in social and biological networks Proc. Natl. Acad. Sci. USA 99, 7821–7826

[18] Bogomolova A, Balk I, Semenov E, Ivaschenko N 2019 Network Analysis of Patenting Trends in Energy Efficiency IOP Conference Series: Earth and Environmental Science 317 https://iopscience.iop.org/article/10.1088/1755-1315/317/1/012005