Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 123 (2018) 271–277

Procedia Computer Science

www.elsevier.com/locate/procedia

8th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2017

Text clustering as graph community detection

Elizaveta K. Mikhina, Vsevolod I. Trifalenkov

National Research Nuclear University "MEPhI" (Moscow Engineering Physics Institute)
Kashirskoe highway 31, 115409, Moscow, Russian Federation
eli-mikhina@yandex.ru

Abstract

This article suggests a method of text clustering that does not depend on any user-set parameters. Text documents and connections between them are represented as graph nodes and edges and graph community detection method is thus applied to the text clustering problem. The method was tested against news articles collections and proved effective – manual and automatic clustering of text documents in collections were same or really close.

Peer-review under responsibility of the scientific committee of the 8th Annual International Conference on Biologically Inspired Cognitive Architectures

Keywords: text clustering, non-parameter clustering, graph community detection, modularity

1 Introduction

At present, processing and analyzing of information often becomes impossible without the use of computer facilities. However, despite the rapidly growing power of computing resources, many tasks are quite difficult to be fully computerized.

There are areas of activity closely related to the analysis of text documents. An example of such documents may news articles that are closely related, but not identical to one another. To computerize the task of analyzing such data, it is necessary to solve the clustering problem, meaning to group data units according to the measure of their similarity.

The existing text clustering algorithms have a significant drawback, namely, the result of clustering depends strongly on a set of parameters. Moreover, in most cases, user can not know in advance how these parameters will affect the final result of the algorithm application. This reduces the computerization potential of such solutions, and also requires that user thoroughly understand both the problem being solved and the mathematical features of the algorithm.

Peer-review under responsibility of the scientific committee of the 8th Annual International Conference on Biologically Inspired Cognitive Architectures
10.1016/j.procs.2018.01.042

2 Text Clustering

Text clustering involves dividing the source collection of texts (documents) into disjoint subsets (clusters) so that each cluster should consist of similar objects, and the objects of different clusters differed significantly among themselves (Manning, 1999).

The initial data for text clustering consists of unstructured texts; texts used as sample collections in experiments described in this article are news articles in Russian.

The task of cluster analysis can be formulated as follows: based on data contained in the matrix of features, split a set of objects (texts) into a number of clusters (subsets) so that each object belongs to one and only one subset. (Nizametdinov, 2012) In this case, close objects belong to the same cluster, and distinct objects belong to different ones.

The concept of proximity and distinction is formally stated by setting a distance function or similarity function.

Based on the matrix of features using the similarity function, a matrix of text mutual similarity is calculated, in which the search is made for maximum values that do not lie on the diagonal of the matrix. The row number and column number of the maximum element correspond to the numbers of the closest documents.

3 Proposed clustering method

The main problem of most clustering algorithms lies in the fact that their result depends heavily on some set of parameters (Kiselev, 2005), and, in most cases, the user can not know in advance how these parameters will affect the final result of the clustering. The key feature of the method proposed by the authors of this article is that it does not depend on any user-set parameters. The decision to assign documents to one cluster is made on the basis of the mutual similarity of the documents with respect to other documents in the collection.

In addition, the proposed method differently estimates the minimum value of the similarity function for different clusters. This allows to select both dense clusters with high minimum value of the similarity function between documents within cluster and sparse clusters with comparatively low value.

The text of the document is considered as a set of words without considering their relative positions. The high-dimensional feature space is the entire set of words used in the document collection, and the non-zero values of the document vector correspond to the words and expressions that occur in it.

To improve the quality of the feature space, the preliminary filtering of the features is performed:

- Stop-words exclusion (built-in stop-words list MSSQL (Microsoft Developer Netbook, 2015));
- Minimal morphological processing (reduction of words to the initial form);
- Evaluation of the importance of the word in the document (based on TF-IDF (Ramos,2003));
- Evaluation of significant bigrams (based on approach described in (Mikhina, 2016)).

As the similarity function authors used the cosine of the angle between a pair of vectors in the multidimensional space corresponding to a pair of documents in the collection.

$$sim^{ij} = \cos(D^i, D^j) = \frac{(D^i \mathrm{x} D^j)}{|D^i| * |D^j|}, \quad (1)$$

where $(D^i \mathrm{x}\, D^j)$ is the scalar product of vectors D^i and D^j, while $|D^i| * |D^j|$ is the product of the lengths of the vectors D^i and D^j.

In the proposed method, text clustering is considered as a particular case of graph community detection (in Russian-language literature the term graph approximation is often attributed to the problem (Il'ev, 2016)). For further clustering, the document collection is represented as a weighted graph

$$G := (V, E), \quad (2)$$

with nodes representing documents

$$V(G) = D, |D| = N \tag{3}$$

and edges representing links between documents (values of the similarity function unequal to 0).

$$E(G) = \{\forall(D^i, D^j): sim(D^i, D^j) > 0\}, \tag{4}$$

Let's denote w_{ij} as the weight of the edge between D^i and D^j:

$$w_{ij} = sim(D^i, D^j) \tag{5}$$

The graph community detection problem can be solved only for sparse graphs, namely, graphs with the number of edges exceeding number of nodes no more than by an order of magnitude (Fortunato, 2010). The graph obtained as a result of processing the text document collection does not meet these requirements, so a method of removing insignificant edges is required.

Thus, the process of clustering a document collection is divided into two stages - representing the collection of documents as a sparse graph and clustering this graph.

Based on the definition of clustering, for each individual document, the problem of clustering can be reduced to the problem of separating documents belonging to the same cluster as the document in question from documents not belonging to that cluster.

Let's consider similarity matrix as a set of N similarity vectors:

$$SIM = \{Sim(D^0), Sim(D^1), \dots, Sim(D^i), \dots, Sim(D^N)\} \tag{6}$$

where each vector describes one document as a set of values of the similarity function between this document and every other document in collection:

$$Sim(D^i) = \{sim^{i1}, sim^{i2}, \dots, sim^{ii}, \dots, sim^{iN}\} \tag{7}$$

Since the value of the similarity function is greater for documents that are closer and less for more distinct ones, for each document D^i there is such a number r_i that all documents with similarity function value between it and the document in question greater than r_i will belong to the same cluster as the document in question, and all documents with similarity function value between it and the document in question less than r_i, will not belong to that cluster.

Based on the assumption that the values of the similarity function between documents belonging to the same cluster will differ slightly, while that between one document belonging to the cluster in question and another document belonging to another cluster will differ considerably, the following approach to finding r_i is suggested:

$$r_i = argmax_{Sim(D^i)_k > 0} \left(\frac{Sim(D^i)_k - Sim(D^i)_{f'(k)}}{Sim(D^i)}\right) \tag{8}$$

where $f'(k)$ returns the number of the next largest element after the considered:

$$f'(k) = argmin_l(Sim^i{}_k - Sim^i{}_l) * \delta'(k, l) \tag{9}$$

$$\delta'(k, l) = \begin{cases} 1, \text{при } (Sim^i{}_k - Sim^i{}_l) > 0 \\ 0, \text{при } (Sim^i{}_k - Sim^i{}_l) \leq 0 \end{cases} \tag{10}$$

The r_i is calculated as the maximum relative change in the similarity function values after sorting them in descending order. I.e. for each element of the similarity vector of the i-th document, greater than 0, find the ratio of the difference between this element and the next largest element to the element itself (the relative change in the value of the similarity function after sorting the values of the similarity functions in descending order). The value of the element corresponding to the maximum relative change will be the desired value.

If the value of the similarity function between documents is less than r_i, the edge between these documents does not affect the clustering result. Hereafter such an edge is referred to as insignificant.

The process of filtering is the transformation of the similarity matrix:

$$sim^{ij} = \frac{1}{2}\left(f(sim^{ij}, i) + f(sim^{ij}, j)\right) \tag{11}$$

$$f(sim^{ij}, i) = \begin{cases} 0, \text{если } sim^{ij} \leq r_i \\ sim^{ij}, \text{если } sim^{ij} > r_i \end{cases} \tag{12}$$

For further clustering of the resuling graph, an agglomeration hierarchical method was applied: the nodes are sequentially joined together in order of decreasing edge weight between them (Nejskij, 2006).

The automatically determined stopping criterion is the change of the cluster quality metric called Modularity (Newman, 2004), denoted by Q:

$$Q = \frac{1}{2m}\sum_{ij}\left(w_{ij} - \frac{(\sum_{k=1}^{m} w_{ik})\, X((\sum_{k=1}^{m} w_{jk}))}{2\sum w}\right)\delta(C_i, C_j) \quad (13)$$

where m is the number of edges, $\sum_{k=1}^{m} w_{ik}$ is the sum of the weights of all edges issuing from the node D^i, $\delta(C_i, C_j)$ is the function that yields one if vertices D^i and D^j are in the same cluster, zero otherwise.

In the clustering process, the change in the quality metric value (ΔQ) after merging each pair of documents was considered. In case of negative value of ΔQ, clustering stopped without merging the documents into a single cluster.

4 Application

To test effectiveness of application of the chosen clustering method a practical experiment was conducted. For this purpose, three collections of news articles from various news sources during same period were collected. Thus, the texts covering the same news topics in different words were collected. Collections were chosen so that they differ in structure as much as possible.

Collection 1 "Uniform distribution" consists of 16 documents divided into four clusters of similar size. Collection 2 "One exception" consists of 16 documents, of which 15 belong to the same cluster and one documents belongs to another one. Collection 3 "One pair" consists of 12 documents, two of them belong to the same cluster while the remaining 10 do not have similar documents.

For result evaluation we use a metric based on ground truth (true clusters manually detected). An averaged F1 (Yang, 2013) was chosen as such metric. In order to adapt the metric to evaluate hierarchical clustering, when evaluating recall, the documents belonging to the child clusters were considered to belong to the parent cluster as well.

Collection 1 "Uniform distribution". Figure 1 depicts the graph representing collection before (a) and after (b) removing insignificant edges.

a) b)

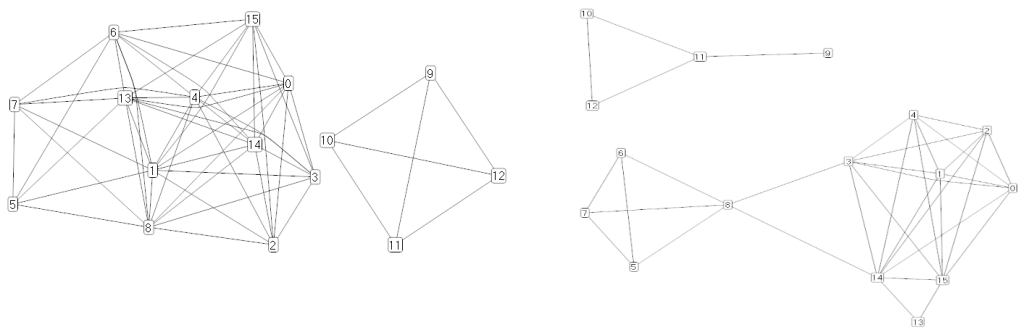

Fig. 1 – Graph representation of document collection 1

After removal of insignificant edges visually the cluster structure becomes more obvious.

Four clusters were manually detected:

$$C_1^* = \{D^0, D^1, D^2, D^3, D^4\}, \quad (14)$$

$$C_2^* = \{D^5, D^6, D^7, D^8\}, \quad (15)$$

(16)

$$C_3^* = \{D^9, D^{10}, D^{11}, D^{12}\}, \quad (17)$$
$$C_4^* = \{D^{13}, D^{14}, D^{15}\}$$

Figure 2 shows the cluster structure detected as a result of the algorithm execution.

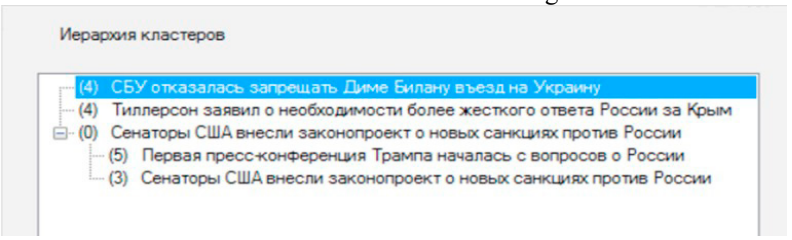

Fig. 2 - Cluster structure detected by the algorithm

Clusters detected manually and by the algorithm are identical. In addition, algorithm detected the relative proximity of clusters $\hat{C}_1$ and $\hat{C}_4$ as they were assigned to the same parent cluster

$$\hat{C}_5 = \{\hat{C}_1, \hat{C}_4\} \quad (18)$$

The averaged F1 is equal to 1.

Collection 2 "One exception". Figure 3 depicts the graph representing collection before (a) and after (b) removing insignificant edges.

a) b)

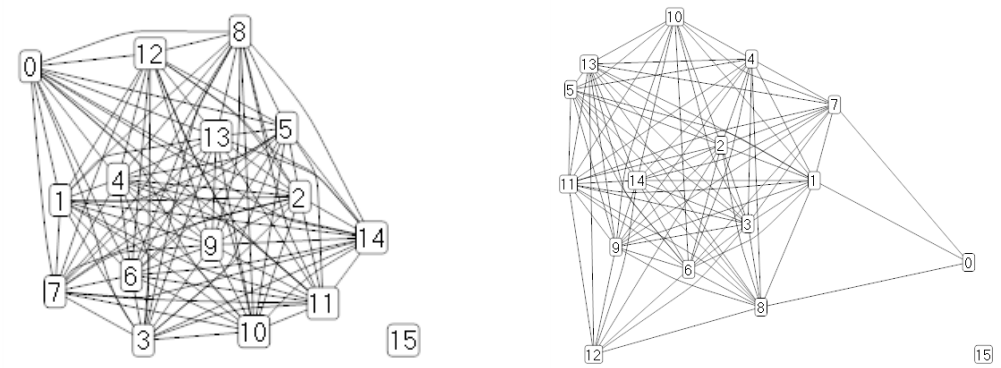

Fig. 3 - Graph representation of document collection 2

After removal of insignificant edges the exclusive proximity of documents 5 and 13 became visually more evident.

Two clusters were detected manually:

$$C_1^* = \{D^0, D^1, D^2, D^3, D^4, D^5, D^6, D^7, D^8, D^9, D^{10}, D^{11}, D^{12}, D^{13}, D^{14}\}, \quad (19)$$
$$C_2^* = \{D^{15}\}, \quad (20)$$

Figure 4 shows the cluster structure detected as a result of the algorithm execution.

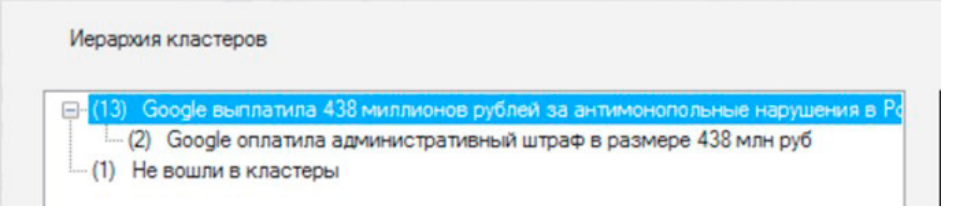

Fig. 4 - Cluster structure detected by the algorithm

Three clusters were detected by algorithm, documents 5 and 13 were put in a separate child cluster.

$$\hat{C}_0 = \{D^5, D^{13}\} \quad (21)$$

$$\hat{C}_1 = \{\hat{C}_0, D^0, D^1, D^2, D^3, D^4, D^6, D^7, D^8, D^9, D^{10}, D^{11}, D^{12}, D^{14}\} \quad (22)$$

$$\hat{C}_2 = \{D^{15}\} \quad (23)$$

The averaged F1 is equal to 0,86.

Collection 3 "One pair". Figure 5 depicts the graph representing collection before (a) and after (b) removing insignificant edges.

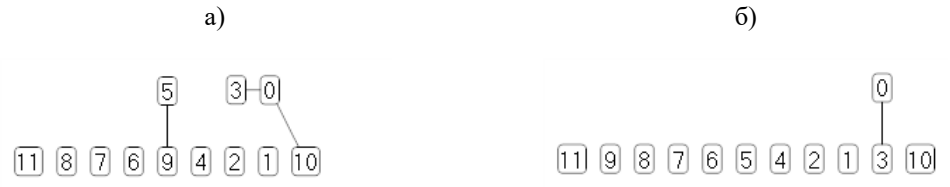

Fig. 5 - Graph representation of document collection 3

After removal of insignificant edges cluster structure of the collection changed.

Eleven clusters were detected manually:

$$C_1^* = \{D^0, D^3\}, \quad (24)$$

$$C_2^* = \{D^1\}, (25) \quad (25)$$

$$C_3^* = \{D^2\}, C_4^* = \{D^4\}, \dots, \quad C_{11}^* = \{D^{11}\} \quad (26)$$

Figure 6 shows the cluster structure detected as a result of the algorithm execution.

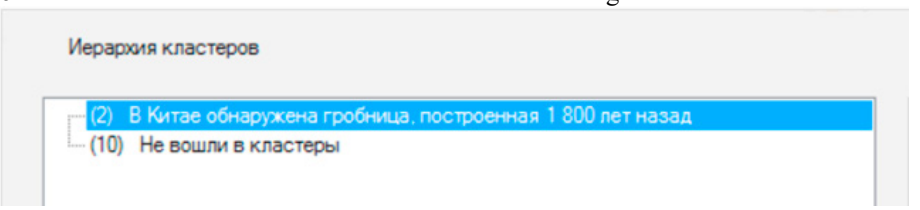

Fig. 6 - Cluster structure detected by the algorithm

Clusters detected manually and by the algorithm are identical. The averaged F1 is equal to 1.

5 Conclusion

This article describes the results of analyzing content from news resources through means of clustering.

Existing text clustering algorithms have a significant drawback - the result of clustering depends strongly on some set of parameters. The proposed text clustering method does not require input parameters.

Based on the method, an algorithm was proposed and implemented as a program in the C# language.

The method proposed and implemented by the authors of the article allows to increase computerization potential of text clustering problem solution, to make clustering result more predictable and to lower the requirements for the knowledge and expertise of the person using the method.

Acknowledgement

This work was supported by the MEPhI Academic Excellence Project (agreement with the Ministry of Education and Science of the Russian Federation of August 27, 2013, project no. 02.a03.21.0005).

References

1. Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing* (Vol. 999). Cambridge: MIT press.Низаметдинов Ш. У. Анализ данных //М.: МИФИ. – 2012. – Т. 286.
2. Nizametdinov Sh. U. Analiz dannyh //M.: MIFI. – 2012. – T. 286.
3. Kiselev, M. V., Pivovarov, V. S., & Shmulevich, M. M. (2005). Metod klasterizacii tekstov, uchityvajushhij sovmestnuju vstrechaemost' kljuchevyh terminov, i ego primenenie k analizu tematicheskoj struktury novostnogo potoka, a takzhe ee dinamiki
4. Microsoft Developer Netbook Stop words and lists of stop words. URL: https://msdn.microsoft.com/ru-ru/library/ms142551(v=sql.100).aspx, 13.06.2017)
5. Ramos, J. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*.
6. Mikhina, E. K., & Trifalenkov, V. I. (2016). METOD AVTOMATIZIROVANNOJ AKTUALIZACII POISKOVYH TEZAURUSOV. Uspehi sovremennoj nauki, 4(11), 99-109.
7. Il'ev, V. P., & Il'eva, S. D. (2016). O zadachah klasterizacii grafov. Vestnik Omskogo universiteta, (2 (80)).
8. Fortunato, S. (2010). Community detection in graphs. *Physics reports*, *486*(3), 75-174.
9. Nejskij, I. M. (2006). Klassifikacija i sravnenie metodov klasterizacii. BBK 32.813 I 76 Sostavitel': JuN Filippovich, 130.
10. Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, *69*(2), 026113.
11. Yang, J., & Leskovec, J. (2013, February). Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining* (pp. 587-596). ACM.