Adaptation of Hierarchical clustering by areas for automatic construction of electronic catalogue

Fedor Vladimirovich Borisuyk
N.I. Lobachevsky State University of Nizhny Novgorod
Nizhny Novgorod, Russia
fedorvb@gmail.com

Vladimir Ivanovich Shvetsov
N.I. Lobachevsky State University of Nizhny Novgorod
Nizhny Novgorod, Russia
shvetsov@unn.ru

Abstract

This paper explores adaptation of Hierarchical clustering by areas algorithm, presented in [1], for automatic construction of electronic catalogue of text documents. Electronic catalogue is traditionally maintained and built by hand, significant research was done to classify the documents to existing hierarchical structures, but little was done about to build the electronic catalogue from scratch. We describe enhancements of Hierarchical clustering by areas algorithm and compare it (to be applied in the field of automatic catalogue construction) with other researches. This paper also proposes an algorithm of feature space reduction for preparation of document representations for text clustering.

Keywords - electronic catalogue, rubrication, informational retrieval, text clustering, area tree

I. INTRODUCTION

At present in different stores of knowledge (electronic and traditional) have accumulated vast amount of information. At the same time because of large volumes of information, their weak structuring and reporting non-electronically, the obtaining of relevant and complete information on a specific topic is quite complex, and majority of the accumulated information resources becomes useless because of it immense. It can be noted that the search of solution for a specific scientific objectives requires high labor costs to find and analyze information on the topic. Therefore, in connection with the stated above, there is the problem of efficient structuring, storing, processing and retrieval of information in the data arrays. To solve this problem people use different thematic classifiers, rubricators, electronic catalogues, which allows finding (either automatically or manually) the documents in a small subset of the document database corresponding to the interesting for the user topic. Electronic catalogues is usually a set of headings grouped into a hierarchy (directory). For each category there is set of documents, which is assigned corresponding to category subject. Currently there are available are two types rubricators - manual and automated. In case of manual rubricator, each new document must be manually analyzed and expert have to define which sections of directory of electronic catalogue it belongs to, after that document becomes available for search. Also there are automated categorization systems, which store sets of characteristics for each category of catalogue and automatically determines corresponding category for the analyzed document. At most the list of attributes for each entry is made by the expert. The disadvantage of the existing automated systems is their static character and the inability to automatically, without the participation of expert rebuild formed earlier catalogue.

In this paper we propose the way of automatic construction of an electronic catalogue for text documents as one of the most effective ways to access the necessary data. The urgency of this problem is increasing as the number and volume of electronic texts is constantly increasing. To automatically construct electronic catalogue the adaptation of novel Hierarchical clustering by areas algorithm is presented. The proposed hierarchical structure of electronic catalogue supposes a quick and efficient way to find the relevant information.

II. RELATED WORK

Much of the previous work concentrates on the classification of text documents, either on flat classification or on hierarchical classification to predefined categories. In case of flat classification of text documents each category treated individually and equally so that no structures exist to define relationships among them [2]. For example, good work in the hierarchical classification to two layer hierarchical structure described by Dumais and Chen in [3], they used SVM classifiers and explored small advantage for accuracy for hierarchical model over flat models.

Only a little research was done in the field of automatic construction of hierarchical catalogue structures. There can be observed two qualitative articles.

Tao Li and Shenghuo Zhu [4] have used linear discriminant projection approach for transformation of document space onto lower-dimensional space and then cluster the documents into hierarchy using Hierarchical agglomerative clustering algorithm [5]. They have found good benefit from using linear projection approach as according to the paper it preserves underlying class structure relation (semantic relations between clustering objects). The paper investigates the affect of using generated hierarchical structure for text classification. They have used LIBSVM [13] as a classifier, which is library for support vector classification, regression and support multi-class

classification. Taken experiments showed that generated hierarchies improve classification performance in most cases, the most significant gain in accuracy (growth up to 53 %) they have reached on the Reuters-top10 collection.

A promising work was done by O. Peskova [6], which presents some improvements for clustering feature selection referred as selective feature space reduction and develops a modification of layerwise clustering method of Ayvazyan [7]. Suggested method of selective feature space reduction decreases feature space in 3.5 times, decreases a computation time, and improves accuracy of clustering algorithm. Presented method was tried on small collections of text documents generated from electronic library on informational technologies http://citforum.ru. Author of the [6] found a 4% advantage in average f-measure of the developed clustering method over Hierarchical agglomerative clustering algorithm [5].

Our work explores the way of automatic construction of an electronic catalogue for text documents as one of the most effective ways to access the necessary data. We propose adaptation of Hierarchical clustering by areas algorithm, which is described in [1].

In next sections we will describe clustering feature selection, description of datasets used for testing of the proposed approach, description of the adapted hierarchical clustering by areas algorithm and evaluation of the proposed approach.

III. CLUSTERING FEATURE SELECTION

For the clustering purpose each text document is presented as the vector of keywords. Universal of document keyword vectors presents keywords feature space. To reduce the feature space stop words (words that usually does not used for search, for example, conjunctions and prepositions) are eliminated. Also modified TFxIDF [8] metric is used to calculate the weight of the word in relation to the document. Finally no more than top 300 features with the highest weight of the word in relation to the document are selected to represent the document. Hereby the following algorithm for keywords extraction from the documents is used:

1) *For all words of the document stem is extracted using Porter algorithm[9]. Number of occurances TF_i of each stem in the document D is counted.*
2) *Stop words are removed from list of extracted words.*
3) *Remove words, which have frequency more than predefined max frequency or less than predefined minimum frequency.*
4) *Weight of the $stem_i$ in the document D is calculated using modified TFxIDF formular, and if to denote - max frequency between all stems as $MaxStemFreq_D$, total number of documents in collection as TDN, number of documents where this stem occurs as DN_i, then we have these formulars to compute weight of the stem:*

$$IDF_i = \log (1+TDN/ DN_i) \quad (1)$$

$$Weight_D(stem_i) = ((0.5+0.5*TF_i)/MaxStemFreq_D)*IDF_i \quad (2)$$

5) *No more than 300 stems with the highest weight are selected as keywords to represent the document (document representation).*

To improve the determinant behavior of features of document representations, we use the idea of Selective feature space reduction presented in [6]. Our implementation is different to [6]. Firstly we cluster the documents collection using modified algorithm of Hierarchical clustering by areas (see section IV.C). Each area in the resulted hierarchical area tree has vector of keywords, which describes it. If to assume areas collection similarly to the documents collection, we can execute keywords extraction algorithm described above on the areas collection to select the vector of most significant keywords of each area in relation to other areas in the areas feature space: this way we reduce the number of keywords used for describing of areas in the area tree (see section IV). Keywords, which are not present in the areas feature space, area removed from document vectors; this way we reduce document vector sizes. The quality of clustering rises two times (for Hierarchical clustering by areas algorithm) in comparison with the clustering without using this technique.

IV. MODIFIED HIERARCHICAL CLUSTERING BY AREAS ALGORITHM

For construction of the electronic catalogue we propose adaptation of Hierarchical clustering by areas algorithm, which is described in [1]. Hierarchical clustering by areas algorithm builds hierarchical tree of the areas, which consists of the documents of initial collection. Characteristics of the areas are calculated during algorithm execution. Final clusters of the algorithm are placed in the nodes of the tree. Node of the hierarchical tree contains objects, which is most closed to each other. Hierarchy of the tree reflects relations between areas.

A. *Initialization of the hierarchical by areas algorithm*

Lets we have incoming stream of the documents, which are supposed to be integrated into the hierarchy. Each document is represented by vector of the keywords. Primarily all incoming documents are put into the recycle bin area of the tree until the number of the documents exceeds a predefined limit, noted as KMax. When the number of the documents of the recycle bin area surpasses KMax, it divides into subareas. Hereby the list of root areas appears.

B. *Improvements of the hierarchical by areas algorithm*

To improve the quality of the hierarchical structure we propose the enhancement of hierarchical by areas algorithm with three additional techniques:

1) *Limitation of depth of the tree.*
2) *Recycle bin at each level.*

Traditionally electronic catalogue has limited number of layers. For example, catalogue of Yandex Corp. has up to 6 layers in depth [10]. Therefore clustering hierarchical algorithm should provide possibility to manage the number of layers in the hierarchy tree. This feature makes the catalogue

to be observable for the user and to locate necessary information easily.

To improve the quality of clustering we introduce the instance of recycle bin at each level. The idea of recycle bin is that all those documents which do not meet entry criteria of the areas of the certain level should be temporary stored in the special area on the same level. Areas on the same level should be as much different from each other as possible, and the idea of recycle bin helps in this question. When the number of objects in the recycle bin surpasses the predefined limit, it is divided and a new detached area is connected to the current level.

C. Phase of processing of incoming document flow

This section describes in detail modified Hierarchical clustering by areas algorithm.

In the capacity of data for the phase of processing there is document, which is represented by vector of keywords, and tree of areas. On the first step of algorithm there is a verification of possibility of correct insertion the document to the area tree. The possibility of correct insertion is determined by measuring of the closeness between document and areas of root level. If closeness does not exceed the dynamically calculated limit, which is defined as minimum of closeness between already processed documents, then document is temporary stored in the recycle bin of the root level. If document has closeness, which is more than predefined limit then, it is directed along the tree to the closest subareas. On the next steps of the algorithm document moves until it meets the closest area of the tree. Document is accommodated in the found area. In case if number of elements of area exceed predefined limit then area is divided into subareas. If number of subareas exceeds the predefined limit then operation of integration of subareas executes. The operation of integration of subareas consists of two basic operations:

1) Partitoining of subareas in two groups of most close to each other.

2) Aggregation of subareas under one area from the elements of the group.

For the notations of modified Hierarchical clustering by areas algorithm see Table I.

TABLE I. NOTATIONS OF THE HIERARCHICAL BY AREAS ALGORITHM

Notation	Description
KMax	Maximal number of elements in the area
RootArea	Root of the tree which points to the list of first level areas of the hierarchical tree
MinProximity	Minimum proximity for the document (to be inserted) that should be between document and node of the area tree.
Divide (area)	Operation of division of area to subareas
proximity(A,B)	Operation of calculation of the nearness between objects A and B
getChildren (area)	Operation of building the list of descendants of area
ConnectToTree (Area)	Connect new area as a child to its parent area from which it was derived (Divide operation) or to the parent of Recycle bin from which was derived.
Integrate (area)	Operation of integration of subareas of the area
RecycleBin	Recycle bin - special area, which stores declined objects.

Take a look at algorithm of insertion of the document in the tree of areas:

1 step. New document Doc has been supplied.

2 step. areaList=getChildren(RootArea);

3 step. FOR EACH area IN areaList: find area which is the most close to Doc.

4 step. Verify if document can be inserted in the hierarchy:

```
IF (proximity (Area, Doc) < MinimumProximity)) {
    RecycleBin.Add (Doc);
    IF (RecycleBin.size() > KMax) {
        Result = Divide (RecycleBin);
        ConnectToTree (Result);
    } End of algorithm;
}
```

5 step. areaList=getChildren(Area);

IF (areaList.size() == 0) GOTO 8 step.

6 step. FOR EACH area IN areaList: find area NArea – which is maximally close to Doc.

7 step.

```
IF (proximity (Area, Doc) <
        proximity (NArea, Doc)) {
    Area = NArea; GOTO 5 step;
} ELSE {
    Area.add (Doc);
    IF (Area.size() > KMax) { divide (Area);
        IF (number of descendants of Area is over limit) {
            Integrate (Area);
        }
    }
    GOTO 8 step.
}
```

8 step. Update vectors of keywords of areas, which is located on the path to the resulted area.

V. RESULTS

For the verification of the presented approach we have used two datasets and compare proposed algorithm with Hierarchical agglomerative clustering algorithm.

A. *Datasets selection*

For testing purposes we have used two datasets. One is subset of 20Newsgroups (20000 articles), which contains 2000 articles evenly divided among 20 Usenet newsgroups (http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html). Second dataset NNSU8 was prepared by us from the scientific articles of Nizhny Novgorod state University, because in future the approach described in this article is supposed to be integrated in the environment of internet portal of Nizhny Novgorod State University as scientific catalogue. NNSU8 contains 1302 scientific articles in 8 scientific areas.

B. *Evaluation*

For evaluation of proposed algorithm of electronic catalogue construction we have used standard external metrics. In the context of clustering tasks, the terms true positives, true negatives, false positives and false negatives are used to compare the given clustering of the documents with the desired, "sample" partitioning of documents to the groups, which is given a priori. This is illustrated by the Table II below:

TABLE II. EVALUATION METRICS

For each pair of the documents D_i and D_j	D_i and D_j contain in one cluster of "sample" partitioning	D_i and D_j contain in different cluster of "sample" partitioning
D_i and D_j contain in one cluster of automatic clustering	tp (true positive)	fp (false positive)
D_i and D_j contains in different clusters of automatic clustering	fn (false negative)	tn (true negative)

We have used these metrics to evaluate the clustering [12]:

- Recall (1) is the fraction of number of correctly grouped documents in automatically generated cluster to the number of documents in "sample" cluster:

$$\text{Recall} = \frac{tp}{tp + fn} \quad (3)$$

- Precision (2) is the fraction of number of correctly grouped documents in automatically generated cluster to the number documents in generated cluster:

$$\text{Precision} = \frac{tp}{tp + fp} \quad (4)$$

- F-measure (3) that combines Precision and Recall is the harmonic mean of precision and recall:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \quad (5)$$

We compared the modified Hierarchical clustering by areas algorithm with Hierarchical agglomerative clustering algorithm (Single link algorithm), implementation of which was taken from the Yooreeka project [11]. The results of computational experiments, which are presented in Table III, show good advantages of Hierarchical clustering by areas algorithm in comparison with Hierarchical agglomerative clustering by means of precision, recall and f-measure characteristics, and computation time. There is a 27% advantage in average f-measure of Hierarchical clustering by areas algorithm over Hierarchical agglomerative clustering algorithm on the NNSU8 collection and a 38% advantage in average f-measure on the 20NewsGroups collection.

TABLE III. EVALUATION OF AVERAGE METRICS OF HIERARCHICAL BY AREAS ALGORITHM IN COMPARISON WITH HIERARCHICAL AGGLOMERATIVE CLUSTERING ALGORITHM

Metric	**Algorithms and datasets**			
	Hierarchical by areas		**Hierarchical Agglomerative**	
	20News groups	***NNSU8***	***20News groups***	***NNSU8***
Recall	0.79	0.66	0.1	0.40
Precision	0.35	0.59	0.11	0.38
F-measure	0.48	0.6	0.1	0.33
Time (msec)	2505	2391	2896	45116

Top levels of catalogue generated by Hierarchical by areas clustering for NNSU8 collection and 20Newsgroups are presented in Table IV and Table V accordingly.

TABLE IV. TOP LEVELS OF CATALOGUE GENERATED BY THE HIERARCHICAL BY AREAS CLUSTERING FOR NNSU8

Areas	**Members**
1	Law, philosophy;
2	Mathematics;
3	Sociology;
4	Economics;
5	Physics
6	Biology, Chemistry;

Addition of new documents to the ready-built catalogue do not need rebuilding of the whole catalogue structure, it supposes the same insertion algorithm, which is presented in section IV.C.

TABLE V. TOP LEVELS OF CATALOGUE GENERATED BY THE HIERARCHICAL BY AREAS CLUSTERING FOR 20NEWSGROUPS

Areas	Members
1	talk.politics.mideast, talk.politics.guns, talk.politics.misc
2	comp.graphics, comp.os.ms-windows.misc
3	rec.sport.baseball, rec.sport.hockey, rec.autos, rec.motorcycles
4	sci.crypt, sci.med, sci.space
5	comp.sys.ibm.pc.hardware; comp.sys.mac.hardware
6	soc.religion.christian
7	sci.electronics; misc.forsale; comp.windows.x
8	talk.religion.misc

CONCLUSION

The research presented in this paper explores adaptation of Hierarchical clustering by areas algorithm for automatic construction of electronic catalogue of text documents. The computational experiments showed advantages of Hierarchical by areas algorithm for automatic electronic catalogue construction in comparison with Hierarchical agglomerative clustering by means of quality and time for computation. In this paper we also present algorithm of feature space reduction for preparation of document representations for clustering, which substantially increases quality of clustering. The addition of new documents to ready-built Hierarchical electronic catalogue, in other words – classification of new documents to the hierarchy, does not need rebuilding of catalogue. Presented in this article approach can be used for construction of web electronic catalogues.

REFERENCES

[1] F.V Borisyuk. and V.I. Shvetsov "New search method based on hierarchical clustering by areas of text documents," Vestnik of N.I. Lobachevsky State University of Nizhny Novgorod, 2009, # 4, pp. 165–171.

[2] Yiming Yang and Xin Liu. "A re-examination of text categorization methods", Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, Berkeley, California, United States, 1999, pp. 42 – 49.

[3] S. T. Dumais and H. Chen (2000). "Hierarchical classification of web content". Proceedings of SIGIR'00, 2000, pp. 256-263.

[4] Tao Li and Shenghuo Zhu. "Hierarcical document classification using automatically generated hierarchy", Journal of Intelligent Information Systems, V. 29 , Issue 2, 2007, pp. 211 - 230.

[5] A.K. Jain and R.C.Dubes, (1988). "Algorithms for clustering data", Prentice Hall, 1988, 320 p.

[6] O.V. Peskova, "Automatic full-text documents classifier building". Electronic Libraries: perspective methods and technology, electronic collections:Proceedings of 9th all-russian scientific conference «RCDL'2008» . Russia, Dubna, 2008, pp. 139-148.

[7] "Applied statistics: Classification and dimintion reduction": Sprav. izd. S.A. Ayvazyan, V.M. Buhshtaber, I.S.Enyukov, L.D. Meshalkin; under the editorship of S.A. Ayvazyan. Moscow: Finance and statistics, 1989. 607 p.

[8] Kelleher D., Luz S. Automatic Hypertext Key phrase Detection // Proceedings of the Nineteenth Inter national Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK. 2005. P. 1608–1610.

[9] The Porter stemming algorithm. http://tartarus.org/~martin/PorterStemmer/

[10] Yandex catalogue: http://help.yandex.ru/catalogue/?id=873432

[11] The Yooreeka project. A library for data mining, machine learning, soft computing, and mathematical analysis. http://code.google.com/p/yooreeka/

[12] Stein, B., S. M. Eissen, F. Wissbrock. On Cluster Validity and the Information Need of Users. In: Proc. 3-rd IASTED Intern. Conf. on Artificial Intelligence and Applications (AIA'03), Acta Press, 2003, pp. 216–221.

[13] Koller D., Sahami M.(1997).Hierarchically classifying documents using very few words. Proceedings of the Fourteenth International Conference on Machine Learning, 1997, pp. 170 – 178.

\