NARROW-DOMAIN SHORT TEXTS CLUSTERING ALGORITHM

Svetlana V. Popova[*], Ivan A. Khodyrev[**]
[*] *Saint-Petersburg State University, Saint-Petersburg State Polytechnic University*
[**] *Saint-Petersburg State Electrotechnical University*
e-mail: spbu@bk.ru, kivan.mih@gmail.com

Abstract

In this paper, we describe the algorithm of narrow-domain short texts clustering, which is based on terms' selection and modification of k-*means* algorithm. Our approach was tested on collections: *CICling–2002* and *SEPLIN-CICling*. Results of tests and conclusions are presented.

Keywords: *information retrieval, texts clustering, narrow-domain short texts clustering, k-means, genetic algorithm.*

1. INTRODUCTION

In the focus of our attention is the task of narrow-domain short texts clustering (we will use shorter term «N-Dst clustering» or «N-Dst» bellow in the article). This research topic is actual now, especially in the field of automated text processing, because of three factors: a practical necessity, difficulty of task, and a small number of research papers. Today most important role in this field is played by Paolo Rosso, Alexander Gelbukh, David Pinto, Mikhail Alexandrov, Marcelo Errecalde, Diego Ingaramo, Leticia C. Cagnina, Fernando Perez-Tellez, John Cardiff and others. Most of these authors conclude that N-Dst clustering problem is difficult, not well researched and there is much work to do [2,7,9].

Results of N-Dst could be used in different ways: searching scientific abstracts, analysis of news articles and any other kind of media sphere, like blogs for example. Clustering abstracts is important to reduce time, spent on search of useful articles in a particular domain. Clustering news about the same topic when the new information about the same event is searched is also a promising task. For example usually in a news flow the same information about the event is repeated in many sources, and new knowledge appears alongside with the old one, thus it could be a challenging task to retrieve really new information from the flow. Proposed technique also could be used for monitoring the reactions and behavioral correlations in social media sphere: blogs, forums, tweets etc. to some influence factor. The factor could be a concrete event or a timed trend of some indicator. One more area where N-Dst could be used is processing of sociological research results (responds, recommendations, and essays on a given topic).

Clustering narrow-domain short texts differs from large texts clustering, because the frequency analysis which is a common technique to work with big texts is not applicable for small ones due to the sparse data.

2. ALGORITHM DESCRIPTION

The goal of an algorithm is to obtain a set of clusters with texts assigned to them, which reflect the topic structure of a source narrow-domain short text collection. At present time basic version of algorithm is developed and implemented. Our approach has two steps:

1. Terms selection and building a set of significant words, which will be used to characterize texts (dimension reduction, collection vocabulary reduction).
2. Clustering, using keyword set from 1.

2.1. Terms Selection

Let $T = \{t_i\}_{i=\overline{1,n}}$ be a set of all words in a collection; $D = \{d_j\}_{j=\overline{1,l}}$ is a set of all texts in a collection. Clustering in n-dimension vector space was taken as a basis. For narrow domain collections words with the highest occur frequency are less significant for clustering, thus they should be filtered. Also the important task is to find words, which reflect the specific features of text groups inside collection. Such words we will call "significant". Usage of significant words helps to reduce the dimension to the size $z<n$, where z is a number of significant words $\Im = \{t_1^{\Im}, t_2^{\Im}, \ldots, t_z^{\Im}\}_{t_i \in T, z<|T|}$. After dimension reduction each text is presented with the binary vector: $t_i^{\Im}:(v_1, v_2, ..., v_{|\Im|})$, where $v_i = \begin{cases} = 1, & t_i \in d_j \\ = 0, & t_i \notin d_j \end{cases}$. The clustering algorithm is a modification of K-means. To calculate distance in k-means we use Euclidean distance.

The choice of significant words is based on three hypothesis.

1. For narrow domain collections we assume that significant words for thematic document groups are not typical for the whole collection, but their placement in texts is near the words, which could be found in the most documents of the collection. This assumption is based on idea that words with high value of *DF* (*Document Frequency*) determine the context of the whole collection and words, which are placed near them, determine the nuances of theirs usage.
2. For short texts we assume that significant word t^1 is often placed together with the word t^2 and rarely far from it. Word t^1 relates to the usage nuances of t^2. This assumption reflects the idea that context-significant groups of words in short texts are often placed together and rarely separately.

3. We assume that semantic of a text group is determined by sets of words, which occur together in the group's texts.

Based on the mentioned assumptions the algorithm which finds significant words was developed. It is divided into two stages, described below. First stage begins with the choice of words with highest value of $DF(t_i) = \sum_{d_j \in D} boolean(t_i \in d_j)$. Then indicator $\vartheta \le \max_i(DF(t_i))$ is selected. Words t^{freq} are selected from the collection's vocabulary with $DF(t^{freq}) \ge \vartheta$. It is better that the number of these words is less than 5. Then information about word pairs is used to create set of words applicants. Using term "word pair" we mean a pair of words, which occur together at least in one text in a window of three words. From the set of all word pairs we choose only "good", meaning of which is described as follows.

Good pairs are chosen with the algorithm: let pair consists of two words t_i and t_j. μ — is a number of pairs (t_i, t_j) inside collection D. Choose a word from a pair with the smallest document frequency rating $DF_{\min}=\min(DF(t_i),DF(t_j))$. If $\mu \ge DF_{\min} - \frac{DF_{\min}}{\alpha}$ then pair (t_i, t_j) is "good". Parameter α is set manually.

We have tested three models of choosing the set of words applicants P.

1. Create set of words, which occur near each t^{freq} in the window of three. For the created set "good" pairs are found. Set of words applicants P consists of words, which are contained in at least one "good" pair.

2. For frequent words t^{freq} the "good" pairs are found (t^{freq} is one of two words in such pairs). Set of words applicants P consists of words, which are contained in at least one "good" pair.

3. Create set of words, which occur near each t^{freq} in the window of three. All these words are considered words applicants and are included into P.

For first tests we used *CICling–2002*[*] collection: worst results were obtained with the third model and best results with the first one. Thus we use first model for an algorithm and all subsequent experiments are made using it.

A subset of collection's words P is an output of the first part. From these words applicants on the second stage, we choose significant words. Other words from collection which are not in P will not be used further.

Second stage involves choice from the set P groups of words with size β, so that all words in each group occur together often in some texts (in two or more). Word, which is placed in at least one such group is considered significant. Words, except significant, will not be used further. Value of β is set manually with respect to the size of P. We use genetic algorithm for this task. Genetic algorithm finds terms, which occur together in documents.

[*] http://sinai.ujaen.es/timm/wiki/index.php/CICLing-2002_Clustering_Corpus

Input parameter β is responsible for the minimal size of individual in population, where each individual is a group of terms gr_k. For example, if β=4, then the result set *GR* will consist of word groups gr_k for which $|gr_k|$=4 and which appear together in *k* texts. We define $GR = \{gr_k\}_{k \in \overline{\{2,|D|\}}}$ and we can select word groups with different *k*. It was done for future research with different sized texts, however for short texts *GR* is taken as a whole.

Basic algorithm is a classical realization of genetic algorithm. It could be presented as a $Gen(W, f_{sel}, m_{mut}, f_{kr}, F_{fit}, F_{fit}^{\max})$, where W — dimension of hypothesis, f_{sel} — selection function, m_{mut} — mutation function, f_{kr} — crossover function, F_{fit} — fitness function, $F_{fit}^{\max}$ — target value for fitness function. *W* is defined with different combinations of words $t_i \in T$, $|W| = 2^{|T|} = \{boolean_i\}_{i=\overline{1,|T|}}$.

On the first step set of individuals is generated: $W_p \subset W$, $W_p = \{w_p\}_{|wp|=\beta}$. Every individual w_p contain β number of terms: $w_p = \{t_i\}_{i \in \{\overline{1,|T|}\}, w_p = \beta}$. Then, for which w_p fitness value F_{fit} is calculated. In current realization F_{fit} is calculated as a number of texts, in which all terms, which create individual, occur $F_{fit}^{\max}=|D|$. Mutation function randomly adds one or two terms to the individual with high value of F_{fit}. Crossover function randomly chooses *m* elements from the individual with high F_{fit} value and replaces them with *m* or *m*+1 elements from another individual with high F_{fit}. Algorithm is iterative one. On each iteration for all individuals of the current population F_{fit} measure is calculated. Selection function f_{sel} chooses only best F_{fit} individuals for the new population from the existing one. Other individuals are replaced with the individuals, obtained with mutation and crossover. Usage of second stage improves results of the method overall.

2.2. Modification of K-means algorithm

K-means algorithm with some modifications was chosen as a basement for clustering. During clustering the optimal number of clusters is usually unknown. Thus we decided to use volatile number of clusters, which is changed during the clustering process. This change is regulated by a number of rules.

1. On the first step algorithm defines one seed c_1 randomly. Then distances ρ to c_1 from all the clustered texts $R=\{\rho_{dj}\}_{dj \in D}$ are calculated. We take the biggest distance $\rho_{\max}$ and determine parameter λ, so that $\lambda \in [\rho_{\max}\text{-}3, \rho_{\max})$ and only few texts $d_j \in D$ should have $\rho \in [\lambda, \rho_{\max}]$.

2. After λ is found, a new seed c_1 is defined randomly and distances ρ_{dj} to this seed are calculated. While iterating the texts, if we find one $d_{g, g \in \{\overline{1,|D|}\}}$ with $\rho_{dg} > \lambda$, then text d_g becomes a new seed c_2.

3. Then new set of distances ρ_{d_j} is calculated, where ρ_{d_j} is a distance of text d_j to the closest seed.

4. Each text, which has $\rho_{d_j} \leq \lambda$ is placed into a cluster, formed by closest seed.

5. If there are texts $d_{q,q\in\{\overline{1,|D|}\}}$ with $\rho_{d_q} > \lambda$, then they are put into temporary set *N*. Next seed is defined as text from *N* with the biggest distance $\rho_{d_i}: c_{next} = \left(d_q^N \mid \rho_{d_q} = \max_{\rho}(N)\right)$. After it algorithm goes to step 3.

If there are no texts $d_{q,q\in\{\overline{1,|D|}\}}$ with $\rho_{d_q} > \lambda$, then k-means' iterative algorithm of seed optimization is initiated. It stops when seeds do not change anymore or when it finds the text $d_{q,q\in\{\overline{1,|D|}\}}$ with $\rho_{d_q} > \lambda$. In last case algorithm goes to step 5.

3. RESULTS OF TESTS

3.1. Test collections

To test algorithms, based on analysis of narrow domain short texts, there exist a number of collections, such as *CICling–2002*, *SEPLIN-CICling*, *Hep-ex*, *KnCr corpus* [10], *EasyAbstracts* and others. Most of them could be found in Internet*. To test quality of clustering we use *FM*-measure based on *F*-measure:

$$FM = \sum_i \frac{G_i}{|D|}, \text{ where } F_{ij} = \frac{2 \cdot P_{ij} \cdot R_{ij}}{P_{ij} + R_{ij}};$$

$$P_{ij} = \frac{|G_i \cap C_j|}{|G_i|}; R_{ij} = \frac{|G_i \cap C_j|}{|C_j|}; G = \{G_i\}_{i=\overline{1,m}}$$

— is an obtained set of clusters, $C = \{C_j\}_{j=\overline{1,n}}$ — set of classes, defined by experts. All test results of this paper are calculated using *FM*-measure. Results of clustering all mentioned collections are present in *FM*-measure, these results are published in [2,4,7]. For experiments we used collections *CICling–2002* and *SEPLN-CICling*. *CICling–2002* contains 48 short texts in the field of linguistics. "Golden standard" contains 4 groups of texts: Linguistic, Ambiguity, Lexicon and Text Processing. *SEPLN-CICling* contains 48 short texts, its "golden standard" contains 4 groups of texts: Morphological-syntactic analysis, Categorization of Documents, Corpus linguistics, Machine translation.

3.2. Parameterization and results

We have conducted a series of experiments, which goal was:

1. Define the relation between parameters α and β;

2. Evaluate the necessity of the second stage from the first part of an algorithm, which precedes clustering.

* http://users.dsic.upv.es/grupos/nle/?file=kop4.php

We used *CICling–2002* and *SEPLN-CICling* collections for these experiments. In each experiment algorithm was started thousand times. Results of clustering on each start were evaluated with *FM*-measure. Based on values of we find three indicators: *FM*max — best *FM*-measure value of the experiment, *FM*min — worst value, *FM*avg — mean value.

For collection *CICling–2002* we used parameter ϑ=29 (ϑ_{max}=30 is a maximum possible value for this collection). Other parameters we defined using the information, that the biggest impact on a result of clustering is made by a number of significant words. This information was obtained by testing. If a number of significant words is about 1% of the whole collection's vocabulary, then the best results will not be reached, but the clustering quality will be still reasonable. If we increase the number of significant words to more than 2,5%, then the best results for *FM*-measure could be received, but the average results become worse. Thus we defined parameters α and β so that the number of significant words lies in between 1% and 2,5% of initial collection's vocabulary. Increase of parameters α and β leads to reduced number of significant words found. It also increases quality of the significant words until their number is not less than 1–1.5% of the vocabulary size. If value of α is big, the importance of parameter β lowers. With big values of both α and β, the result of significant words allocation is absent. We used β=2 and β=3 for *CICling–2002* collections. Results of testing with different values of parameter α are presented in a Table 1 (* — number of significant words). It also contains the projection of genetic algorithm usage. For *SEPLN-CICling* we use β=2 and β=3, α=5 and α=4, ϑ=26 and ϑ=18 (ϑ_{max}=27 is a maximum possible value for this collection). We compare test results with results that were presented in another work [2] for *CICling–2002* in Table 2 and for *SEPLN-CICling* in Table 3 (K-Means [3], MajorClust [13], DBSCAN [6], CLUDIPSO [2]). We also compared results with the case when instead of genetic algorithm, precise method was used. It chooses all pairs and triplets of words, which occur together in more than one text. Set of significant words is built as an union of all obtained pairs, triplets. Result of algorithm's work with precise method is shown in the Table 4.

Usage of modified k-means algorithm is possible because the number of words which present text vectors is small. Thus during first step of clustering distances from the seed to the most distant texts have small variations (as example: from 9 to 11) and the value of parameter λ may be automatically selected. We recommend to defined λ by using one of the highest values of distances, obtained during first step of clustering; the number of texts with

Table 1. Best FM-measures for different values of parameter α

		α=1			α=4			α=5		
		Fm (*avg*)	*Fm* (*min*)	*Fm* (*max*)	*Fm* (*avg*)	*Fm* (*min*)	*Fm* (*max*)	*Fm* (*avg*)	*Fm* (*min*)	*Fm* (*max*)
*		110			67			36		
No GA	*avg*	**0,51**	0,39	0,54	**0,49**	0,34	0,57	**0,47**	0,4	0,59
	min	0,45	**0,36**	0,59	0,47	**0,39**	0,55	0,47	**0,4**	0,59
	max	0,45	0,36	**0,59**	0,45	0,34	**0,6**	0,47	0,4	**0,59**
*										
With GA	*avg*	**0,51**	0,39	0,54	**0,5**	0,4	0,62	**0,5**	0,4	0,58
	min	0,51	**0,39**	0,54	0,5	**0,4**	0,62	0,49	**0,4**	0,63
	max	0,45	0,33	**0,65**	0,49	0,36	**0,66**	0,46	0,38	**0,66**
		α=6			α=7			α=8		
		Fm (*avg*)	*Fm* (*min*)	*Fm* (*max*)	*Fm* (*avg*)	*Fm* (*min*)	*Fm* (*max*)	*Fm* (*avg*)	*Fm* (*min*)	*Fm* (*max*)
*		26			15			10		
No GA	*avg*	**0,48**	0,38	0,61	0,49	0,42	0,6	**0,49**	0,42	0,6
	min	0,48	**0,38**	0,54	0,47	**0,43**	0,61	0,49	**0,42**	0,6
	max	0,48	0,38	**0,61**	0,47	0,43	**0,61**	0,49	0,42	**0,6**
*		10–14			8					
With GA	*avg*	**0,51**	0,4	0,6	0,48	0,44	0,51	-	-	-
	min	0,48	**0,41**	0,56	0,48	0,44	0,51	-	-	-
	max	0,51	0,4	**0,61**	0,44	0,35	0,56	-	-	-

Table 2 (Part 1). *CICling–2002*: best *FM*-measures for different algorithms

	Fm (*avg*)	*Fm* (*min*)	*Fm* (*max*)
K-Means	0,45	0,35	0,6
MajorClust	0,43	0,37	0,58
DBSCAN	0,47	0,42	0,56
CLUDIPSO	0,63	0,42	0,74

Table 2 (Part 2). *CICling–2002*: best *FM*-measures for algorithm described here

	Fm (*avg*)	*Fm* (*min*)	*Fm* (*max*)
Fm (*avg*)	**0,51**	0,4	0,6
Fm (*min*)	0,49	**0,42**	0,6
Fm (*max*)	0,49	0,36	**0,66**

Table 3 (Part 1). *SEPLN-CICling*: best *FM*-measures for different algorithms

	Fm (*avg*)	*Fm* (*min*)	*Fm* (*max*)
K-Means	0,49	0,36	0,69
MajorClust	0,59	0,4	0,77
DBSCAN	0,63	0,4	0,77
CLUDIPSO	0,72	0,58	0,85

Table 3 (Part 2). *SEPLN-CICling*: best *FM*-measures for algorithm described here

	Fm (*avg*)	*Fm* (*min*)	*Fm* (*max*)
Fm (*avg*)	**0,6**	0,45	0,71
Fm (*min*)	0,58	**0,52**	0,67
Fm (*max*)	0,6	0,45	**0,71**

Table 4. *CICling–2002*: best *FM*-measures for algorithm with precise method

	Fm (*avg*)	*Fm* (*min*)	*Fm* (*max*)
Fm (*avg*)	**0,49**	0,4	0,61
Fm (*min*)	0,49	**0,4**	0,61
Fm (*max*)	0,49	0,4	**0,61**

distances higher than λ shouldn't exceed 7 (for these collections). Getting a small variation between the highest distances is possible if a small number of words for presenting texts vectors is use. Binary vectors that present texts shouldn't have much "0" ("0" in text's vector means that this text doesn't contain concrete word). The first part (terms selection) of this algorithm filters out words with a low *DF*. Words that are selected in the first part of algorithm are context words, not just common. This is a result of a good pairs selection. However, during this selection some words with a single appearance in some of collection's texts could be added to a resulting set of words. This problem is solved with GA, which filters them. Thus, words that have a specific context and high *DF* are selected. This provides the result of the algorithm.

4. CONCLUSION

We assume, that initially adopted hypothesis move us in the right direction, because the set of significant words, which is built as a result of first stage of an algorithm is informative and contains terms, which reflect the nuances of the texts. There are some examples of significant words here for *CICling–2002*: base, corpu, lexic, select, paper, evalu, languag, word, larg, document, ap-

proach, differ, linguist, inform, kind, knowledg, mean, automat, system; and for *SEPLING–CICling*: clustering, based, linguistic, language, corpus, order, translation, important, computational, part, results, machine.

Genetic algorithm chooses about 50–70% of words, found by precise method. Despite this, the clustering algorithm with the output of genetic algorithm gives better results. We assume that better results are obtained because GA uses random choice of objects to process. Probability that genetic algorithm will create a pair or triplet with some word rises with the rise of the probability that this word occur together with another word more than in one text. In other words GA filters random words, which occur in different texts without dependency to other terms, thus we can say that terms without specific context of usage will be filtered by GA.

Proposed modification of k-means gives better results comparing with the non-modified version, but we believe that the clustering algorithm needs improvement. This is due to the specific features of narrow domain collections. When clustering narrow domain collections, most documents in the clustering area are placed very close to each other and even if there are 2 or 3 defined seeds with large relative distance between them, the border between clusters, which divides dense area of texts is more or less illusory. We assume that to solve this problem, more complex algorithms, which measure increase and decrease of objects' densities in the clustering area, are required.

Words with low value of *DF* don't have enough significance for clustering and could be neglected, because the algorithm works with high *DF* value words. We also assume that context significant words are characterized by high *DF* value for such collections. We believe this is an interesting observation that requires further research and could lead to simplification and improvement of terms' selection procedure.

REFERENCES

1. **Alexandrov M., Gelbukh A., and Rosso P.** 2005. An approach to clustering abstracts. In Proceedings of the 10th International NLDB-05 Conference, volume 3513 of Lecture Notes in Computer Science, pages 8–13. Springer-Verlag.
2. **Cagnina, L., Errecalde, M., Ingaramo, D., Rosso, P.**: A discrete particle swarm optimizer for clustering short-text corpora. In: BIOMA08, pp. 93–103 (2008)
3. **Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze**, Introduction to Information Retrieval, Cambridge University Press. 2008. DOI= http://nlp.stanford.edu/IR-book/information-retrieval-book.html
4. **Errecalde M., Ingaramo D., Rosso P.** 2010. A new AntTree-based algorithm for clustering short-text corpora.
5. DOI= http://users.dsic.upv.es/~prosso/resources/ErrecaldeEtAl_JCST10.pdf

6. **Errecalde M., Ingaramo D.** 2008. Short-text corpora for clustering evaluation. Technical report, LIDIC.
7. DOI= http://www.dirinfo.unsl.edu.ar/~ia/resources/shortexts.pdf
8. **M. Ester, H. Kriegel, J. Sander and X. Xu.** A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proc. of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), pages 226–231. 1996. DOI= http://ifsc.ualr.edu/xwxu/publications/kdd-96.pdf
9. **Ingaramo D., Pinto D., Rosso P., and Errecalde M.** 2008. Evaluation of internal validity measures in short-text corpora. In Proc. of the CICLing 2008 Conference, volume 4919 of Lecture Notes in Computer Science, Springer-Verlag, pages 555–567.
10. **Makagonov P., Alexandrov M., and Gelbukh A.** 2004. Clustering abstracts instead of full texts. In Proc. of the Text, Speech and Dialogue 2004 Conference - TSD04, volume 3206 of Lecture Notes in Artificial Intelligence, pages 129–135. Springer-Verlag.
11. **Pinto D.** 2007. Analysis of narrow-domain short texts clustering. Research report for "Diploma de Estudios Avanzados (DEA)", Department of Information Systems and Computation, UPV.
12. **Pinto D., Jimenez-Salazar H., and Rosso P.** 2006. Clustering abstracts of scientific texts using the transition point technique. In Proc. of the CICLing 2006 Conference, volume 3878 of Lecture Notes in Computer Science, pages 536–546. Springer-Verlag.
13. **Pinto D. and Rosso P.** 2007. On the relative hardness of clustering corpora. In Proc. of the Text, Speech and Dialogue 2007 Conference - TSD07, volume 4629 of Lecture Notes in Artificial Intelligence, pages 155–161. Springer-Verlag.
14. **Pinto D. and Rosso P.** 2006. KnCr: A short-text narrow-domain sub-corpus of Medline. In Proc. of TLH 2006 Conference, Advances in Computer Science, pages 266–269.
15. **Stein, B., O. Niggemann.** On the Nature of Structure and its Identification. In: Graph Theoretic Concepts in Computer Science. LNCS, N 1665, Springer, 1999, pp.122–134.