A Particle Swarm Optimizer to cluster short-text corpora: a performance study

Diego A. Ingaramo[1], Leticia C. Cagnina[1], Marcelo L. Errecalde[1], and Paolo Rosso[2⋆]

[1] LIDIC (Research Group). Universidad Nacional de San Luis. San Luis, Argentina. {daingara,lcagnina,merreca}@unsl.edu.ar

[2] Natural Language Engineering Lab. - ELiRF, DSIC, Universidad Politécnica de Valencia. Valencia, España. prosso@dsic.upv.es

Abstract. Short-text clustering is currently an important research area because of its applicability to web information retrieval, text generation and text mining. Some previous works have demonstrated the effectiveness of a discrete Particle Swarm Optimizer algorithm, named CLUDIPSO, for clustering corpora containing very short documents. In these studies, CLUDIPSO was evaluated with small collections and, in all the considered cases, it outperformed the performance of algorithms representative of the state-of-the-art in the area. An interesting aspect to consider with CLUDIPSO (and other bio-inspired methods) is how well it can scale up to larger (more realistic) corpora. This paper presents a preliminary study showing the performance of CLUDIPSO on short-text corpora of different sizes. The results were compared with those of the most effective clustering algorithms in the area. The experimental work gives strong evidence about the effectiveness of CLUDIPSO on small collections and some drawbacks of this algorithm to manage larger collections. With respect to this last aspect, some possible reasons of the poor behavior of CLUDIPSO in these cases is discussed and the current work to solve this weakness is briefly described.

1 Introduction

In a document clustering problem, the main goal is to group a set of documents into different clusters. In this context, the clustering of short-text corpora, is one of the most difficult tasks in natural language processing due to the low frequencies of terms in the documents.

In document clustering, the information about categories and correctly categorized documents is not provided in advance. An important consequence of this lack of information is that in realistic document clustering problems, results can not usually be evaluated with typical *external* measures like F-Measure and the Entropy, because the correct categorizations specified by a human expert are not available. Therefore, the quality of the resulting groups is evaluated with respect

⋆ The research work of the last two authors is partially funded by the MICINN project TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 (Plan I+D+i).

to *structural* properties expressed in different *Internal Clustering Validity Measures* (ICVMs). Classical ICVMs used as cluster validity measures include the Dunn and Davies-Bouldin indexes, the *Global Silhouette* (GS) coefficient and, new graph-based measures such as the *Expected Density Measure* (EDM) and the λ-Measure (see [7] for detailed descriptions of these ICVMs).

The use of these unsupervised measures of cluster validity -or any arbitrary criterion function that gives a reasonable estimation of the quality of the obtained groups- is not limited to the cluster evaluation stage. They can also be used as an *objective function* that the clustering algorithm attempts to optimize *during* the grouping process. This approach has been adopted by clustering algorithms like CLUDIPSO, a discrete Particle Swarm Optimizer (PSO) which obtained in previous works [6] interesting results on small short-text collections, using the GS coefficient as objective function.

This paper reports some works in progress related to the performance of CLUDIPSO on short-text corpora of different sizes. The aim of this investigation is to detect possible limitations of this algorithm to scale up to larger corpora than those considered in the initial studies. In order to analyze this aspect, CLUDIPSO was compared with some of the most effective clustering algorithms in the area and with a representative number of corpora of different sizes. The experimental work confirmed the good performance of CLUDIPSO on small collections, but it also showed some limitations to deal with larger collections. The present work poses some possible reasons of the poor behavior of CLUDIPSO in these cases and also describes some works that are currently being developed in order to improve the CLUDIPSO performance on larger collections.

The remainder of the paper is organized as follows. Section 2 describes CLUDIPSO, the PSO-based algorithm under study. Section 3 describes some general features of the corpora used in the experiments. The experimental setup and the analysis of the results obtained from the empirical study is provided in Section 4. Finally, some general conclusions are drawn and present and future work is discussed in Section 5.

2 The CLUDIPSO Algorithm

In a previous work [6] a discrete PSO algorithm named CLUDIPSO (CLUstering with a DIscrete PSO) was presented. The main characteristics of CLUDIPSO are briefly listed below:[3]

- Each valid clustering is represented as a *particle*. The particles are n-dimensional integer vectors, where n = number of documents in the collection.
- The best position found so far for the swarm (*gbest*) and the best position reached by each particle (*pbest*) are recorded.
- Velocity updating formula: $v_{id} = w(v_{id} + \gamma_1(pb_{id} - par_{id}) + \gamma_2(pg_d - par_{id}))$. Where par_{id} is the value of the particle i at the dimension d, v_{id} is the velocity of particle i at the dimension d, w is the inertia factor [3] whose goal is to

[3] For a more detailed description see [6].

balance global exploration and local exploitation, γ_1 is the personal learning factor, and γ_2 the social learning factor, both multiplied by 2 different random numbers within the range [0,1]. pb_{id} is the best position reached by the particle i and pg_d is the best position reached by any particle in the swarm.
- Position updating formula: $par_{id} = pb_{id}$ proposed in [6] for discrete versions of PSO.
- Dynamic mutation operator [2] applied with a pm-probability calculated with the total number of iterations in the algorithm ($cycles$) and the current cycle number: $pm = max\_pm - \frac{max\_pm - min\_pm}{max\_cycle} * current\_cycle$. Where max_pm and min_pm are the maximum and minimum values that pm can take, max_cycle is the total number of cycles and the current cycle in the iterative process is $current\_cycle$. The mutation operation is applied if the particle is the same that its own $pbest$, as was suggest by [5]. The mutation operator swaps two random dimensions of the particle.

3 Data Sets

The inherent difficulty of short-document clustering problems requires a detailed analysis of the features of each collection used in the experiments. For this reason, some specific characteristics such as document length and total number of terms are considered below.

In this experimental work the Micro4News, EasyAbstracts, SEPLN-CICLING and CICling-2002 short-text corpora were selected. These are considered small collections because they only have 48 documents. Several works [9, 1, 10, 7] have used these collections to test the performance of their approaches and the interested reader can obtain more information about them in [4].

Other three collections (with different characteristics) were also considered: R4, R6 and R8B which are subsets of the well known R8-Test corpus, a subcollection of the Reuters-21578 dataset. The main differences between the three corpora are the total number of documents, terms and groups. The detailed descriptions of these collections are presented in Tables 1, 2 and 3, respectively. The Reuters-derived collections are considerably larger than the four short-text corpora mentioned above.

Table 1. Main characteristics of the R4 corpus

Category	**# docs**
trade	102
grain	34
interest	87
ship	43

Feature	**Value**
Size of the corpus (KBytes)	184
# categories	4
# tot. docs	266
# tot. terms	27623
Voc. size	4578
Term average per document	166.4

Table 2. Main characteristics of the R6 corpus

Category	# docs
trade	102
grain	34
monex-fx	130
crude	140
interest	87
ship	43

Feature	Value
Size of the corpus (KBytes)	313
# categories	6
# tot. docs	536
# tot. terms	53494
Voc. size	4600
Term average per document	99.8

Table 3. Main characteristics of the R8B corpus

Category	# docs
trade	102
grain	34
monex-fx	130
crude	140
interest	87
acq	140
ship	43
earn	140

Feature	Value
Size of the corpus (KBytes)	415
# categories	8
# tot. docs	816
# tot. terms	71842
Voc. size	5854
Term average per document	88.04

4 Experimental Results

In the experiments, 50 independent runs per problem were performed, with 10,000 iterations (*cycles*) per run. CLUDIPSO used the following parameters: swarm size = 50 particles, dimensions at each particle = number of documents (N), $pm\_min = 0.4$, $pm\_max = 0.9$, inertia factor $w = 0.9$, personal and social learning factors for γ_1 and γ_2 were set to 1.0. The parameter settings such as swarm size, mutation probability and learning factors were empirically derived after numerous experiments. It is important to note that for big collections as Reuters-derived ones, the algorithm was tested with more iterations and more particles. CLUDIPSO obtained with those settings the best value in the last cycles but the improvements were not significant compared to the increase in the execution time of a single run. The objective function to be optimized was GS.

The GS measure combines two key aspects to determine the quality of a given clustering: *cohesion* and *separation*. Cohesion measures how closely related are

objects in a cluster whereas separation quantifies how distinct (well-separated) a cluster from another is. The GS coefficient of a clustering is the average cluster silhouette of all the obtained groups. The cluster silhouette of a cluster C also is an average silhouette coefficient but, in this case, of all objects belonging to C. Therefore, the fundamental component of this measure is the formula used for determining the silhouette coefficient of any arbitrary object i, that we will refer as $s(i)$ and is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

with $-1 \leq s(i) \leq 1$. The $a(i)$ value denotes the average dissimilarity of the object i to the remaining objects in its own cluster, and $b(i)$ is the average dissimilarity of the object i to all objects in the nearest cluster. From this formula it can be observed that negative values for this measure are undesirable and values close to 1 are the best.

The results were compared with those obtained with other three clustering algorithms: K-means, K-MajorClust [11] and CHAMELEON [8]. K-means is one of the most popular clustering algorithms and, K-MajorClust and CHAMELEON are representative of the density-based and graph-based approaches to the clustering problem. Information about the correct number of groups (k) has to be provided to the algorithms.

The quality of the results was evaluated using the classical (external) F-measure on the clusterings that each algorithm generated in 50 independent runs per collection. The reported results correspond to the minimum (F_{min}), maximum (F_{max}) and average (F_{avg}) F-measure values. Tables 4 and 5 show the F_{avg}, F_{min} and F_{max} values obtained with the seven collections. The values highlighted in bold indicate the best results obtained.

	Micro4News			EasyAbstracts			SEPLN-CICLing			CICling-2002		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means	0.67	0.41	0.96	0.54	0.31	0.71	0.49	0.36	0.69	0.45	0.35	0.6
K-MajorClust	**0.95**	**0.94**	0.96	0.71	0.48	**0.98**	0.63	0.52	0.75	0.39	0.36	0.48
CHAMELEON	0.76	0.46	0.96	0.74	0.39	0.96	0.64	0.4	0.76	0.46	0.38	0.52
CLUDIPSO	0.93	0.85	**1**	**0.92**	**0.85**	**0.98**	**0.72**	**0.58**	**0.85**	**0.6**	**0.47**	**0.73**

Table 4. F-measures values per collection.

With the small collections (less than 50 documents) it is observed in Table 4 that CLUDIPSO obtained the best F_{max} values and, in some cases, with a significative difference with respect to the other tested algorithms (see for instance, the results with SEPLN-CICLING and CICling-2002). Similar results can be observed with the F_{min} and F_{avg} values in collections as EasyAbstracts, SEPLN-CICLING and CICling-2002 in which the minimum and averaged values of CLUDIPSO clearly outperformed those of the remainder algorithms. The highest possible

value of F_{max} (which is 1 and means the perfect classification) was reached by CLUDIPSO with Micro4News although the best F_{min} and F_{avg} values for this collection were obtained by K-MajorClust. These results are conclusive with respect to the good performance that CLUDIPSO can obtain with small short-text collections with very different characteristics.

	R4			R6			R8B		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means	**0.73**	**0.57**	**0.91**	**0.63**	**0.51**	**0.81**	**0.64**	0.48	**0.78**
K-MajorClust	0.70	0.45	0.79	0.53	0.36	0.74	0.5	0.28	0.68
CHAMELEON	0.61	0.47	0.83	0.52	0.42	0.66	0.62	**0.57**	0.71
CLUDIPSO	0.64	0.48	0.75	0.31	0.26	0.38	0.21	0.18	0.25

Table 5. Best F-measures values per collection.

For the Reuters-derived collections, CLUDIPSO obtained very poor results in all cases, being K-Means the algorithm that achieved the best results in almost all collections with more than 50 documents. The very different performance of CLUDIPSO in both kinds of collections (that is with few and many documents) is probably due to the difficulty of the algorithm to explore the big search space that Reuters corpora imply. This could be observed in the little improving of performance during the execution of CLUDIPSO when it had to evolve the big size particles (one dimension for each document) for the Reuters collections. Additionally, the mechanism used to update the particles (proposed for discrete versions of PSO in [6]) causes a slow search space exploration making the algorithm unable to find good solutions in a considerable amount of cycles (that is 10,000). This slow exploration can be observed when CLUDIPSO finishes the run and the last performance improvement is obtained in the last iterations of the algorithm.

Additional information on the bad behavior of CLUDIPSO with the Reuters collections can be obtained from the Boxplots with the distribution of F-Measure values (averaged) shown in Figure 1. [4]

In Figure 1 (top), the results obtained by CLUDIPSO and K-Means with R4 showed a significative dispersion. This means that both algorithms did not obtain similar results in the total of runs done. The median value in the boxplot of CLUDIPSO presents a strong bias to the right side showing that many of the best values in all runs are around 0.65. The median value of K-Means is better than that obtained with CLUDIPSO (around 0.7) and K-MajorClust does not evidence a big dispersion but all values in all runs are lower than those of CLUDIPSO and K-Means. Then, studying the distribution of averaged F-Measure values, the boxplots do not show a big difference of performance between

[4] CHAMELEON is not considered in the boxplots for R4 and R6 corpora because it obtains lower values than the other methods making the results incomparable.

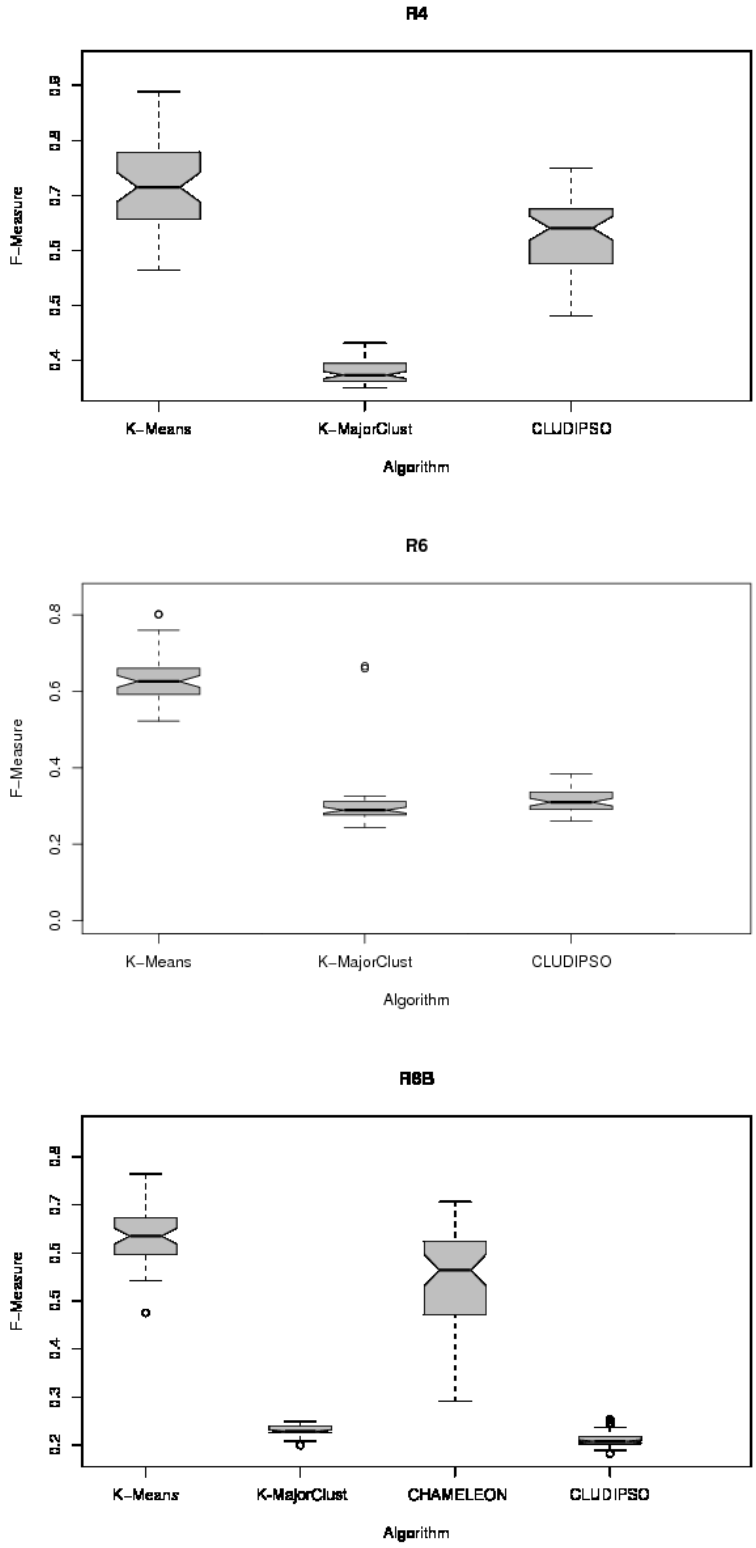

Fig. 1. Boxplots for the Reuters collections: R4 (top), R6 (middle) and R8B (bottom)

CLUDIPSO and K-Means although the last algorithm obviously outperforms the first.

With R6, a collection bigger than R4, the Figure 1 (middle) shows that CLUDIPSO gets similar results to K-MajorClust, with similar median values and low dispersions in their boxplots. The boxplot of K-Means shows the best median value (around 0.6) but with a high dispersion of values. Again for this collection, K-Means outperforms CLUDIPSO but the differences in favor of K-Means tend to increase with respect to the previous collection (R4).

Figure 1 (bottom) shows that, for the R8B collection, CLUDIPSO and K-MajorClust have a very low performance (median values around 0.25) but they have a better dispersion than K-Means and CHAMELEON. Obviously, the best median values of K-Means and CHAMELEON are conclusive about a better performance of these algorithms with respect to CLUDIPSO.

As final conclusion of this statistical distribution study, it is possible to state that, according the search space grows (the number of documents increases), CLUDIPSO can not converge into good quality results even though it can still be considered a "robust" algorithm observing the dispersion of its results.

5 Conclusions and Future Work

This work presented a study of performance of CLUDIPSO, a novel PSO-based clustering algorithm. The results obtained by CLUDIPSO indicate that the approach is a highly competitive alternative to solve problems of short-text corpora clustering, with very small collections of no more than 50 documents. In this work, CLUDIPSO was also tested with larger size collections and the performance was not comparable with other traditional algorithms like K-means. In these comparisons, a constant deterioration of the F-measure values obtained with CLUDIPSO was observed while the number of documents in the collections was increased.

Future works include the modification in the representation of the particles to consider sub-groups of documents (that is, reduce the length of the particle representation) and the adaptation of several stages of the CLUDIPSO algorithm to incorporate information about the clustering problem itself (that is, reduce the result search space). Also, the mechanism to update the particles should be improved in order to accelerate the exploration of the search space. A reliability of the obtained results would also be an aspect to be considered in future works.

References

1. M. Alexandrov, A. Gelbukh, and P. Rosso. An approach to clustering abstracts. *Proc. of the 10th Int. NLDB-05 Conference, LNCS*, 3513:8–13, 2005. Springer-Verlag.
2. L. Cagnina, S. Esquivel, and R. Gallard. Particle swarm optimization for sequencing problems: a case study. In *Congress on Evolutionary Computation*, pages 536–541, 2004.

3. R. Eberhart and Y. Shi. A modified particle swarm optimizer. In *International Conference on Evolutionary Computation*. IEEE Service Center, 1998.
4. M. Errecalde and D. Ingaramo. Short-text corpora for clustering evaluation. Technical report, LIDIC, 2008.
5. X. Hu, R. Eberhart, and Y. Shi. Swarm intelligence for permutation optimization: a case study on n-queens problem. In *Proc. of the IEEE Swarm Intelligence Symposium*, pages 243–246, 2003.
6. D. Ingaramo, M. Errecalde, L. Cagnina, and P. Rosso. *Computational Intelligence and Bioengineering*, chapter Particle Swarm Optimization for Clustering short-text Corpora, pages 3–19. IOS Press, 2009. F. Masulli et al. (Eds.).
7. D. Ingaramo, D. Pinto, P. Rosso, and M. Errecalde. Evaluation of internal validity measures in short-text corpora. *Proc. of the CICLing 2008 Conference. LNCS*, 4919:555–567, 2008. Publisher Springer-Verlag.
8. G. Karypis, E. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32:68–75, 1999.
9. P. Makagonov, M. Alexandrov, and A. Gelbukh. Clustering abstracts instead of full texts. In *Proc. of TSD-2004*, volume 3206 of *LNAI*, pages 129–135. Springer-Verlag, 2004.
10. D. Pinto, J. Benedí, and P. Rosso. Clustering narrow-domain short texts by using the Kullback-Leibler distance. *Proc. of the CICLing 2007 Conference, LNCS*, 4394:611–622, 2007. Springer-Verlag.
11. B. Stein and O. Niggemann. On the Nature of Structure and its Identification. *Proc. of the 25th International Workshop on Graph Theoretic Concepts in Computer Science - WG99. LNCS*, 1665:122–134, 1999. Springer-Verlag.