Cagnina, L.; Errecalde, M.; Ingaramo, D.; Rosso, P. (2014). An efficient Particle Swarm Optimization approach to cluster short texts. Information Sciences. 265:36-49. doi:10.1016/j.ins.2013.12.010.

An efficient Particle Swarm Optimization approach to cluster short texts

Leticia Cagnina[1], Marcelo Errecalde[1], Diego Ingaramo[1] and Paolo Rosso[2]

[1] *LIDIC (Research Group) Universidad Nacional de San Luis, Argentina.*
{lcagnina,merreca}@unsl.edu.ar
[2] *Natural Language Engineering Lab. - ELiRF, DSIC, Universitat Politècnica de València, Spain. prosso@dsic.upv.es*

Abstract

Short texts such as evaluations of commercial products, news, FAQ's and scientific abstracts are important resources on the Web due to the constant requirements of people to use this on line information in real life. In this context, the clustering of short texts is a significant analysis task and a discrete Particle Swarm Optimization (PSO) algorithm named CLUDIPSO has recently shown a promising performance in this type of problems. CLUDIPSO obtained high quality results with small corpora although, with larger corpora, a significant deterioration of performance was observed. This article presents CLUDIPSO*, an improved version of CLUDIPSO, which includes a different representation of particles, a more efficient evaluation of the function to be optimized and some modifications in the mutation operator. Experimental results with corpora containing scientific abstracts, news and short legal documents obtained from the Web, show that CLUDIPSO* is an effective clustering method for short-text corpora of small and medium size.

Keywords: Clustering, Short-text Corpora, Particle Swarm Optimization

1. Introduction

In recent years, document clustering has become a fundamental process in many tasks as enhancing the results returned by search engines, text mining, unsupervised text organization and information retrieval. In many of these domains, the clustering task has involved documents and content available on the Web. This interest in the use of clustering techniques in these cases, can

be appreciated in past events such as the *The Spock Challenge* competition, but also in more recent events related to benchmarking activities on *Web People Search*[1].

In this context, much of the useful information to be processed is taken from Web repositories whose documents are, frequently, short texts with a few tens or hundreds words, such as scientific abstracts, news and short technical and legal documents. For instance, in most digital libraries and on line repositories users have usually free access to abstracts of scientific papers but not to their full texts. Organizing that huge volume of short texts is an important challenge, as it has been observed in many works on clustering of scientific abstracts [1, 10, 34].

Several techniques have been developed to solve clustering problems and those based on the Swarm Intelligence (SI) paradigm seem to be specially attractive because of their robust performance [4, 30, 31, 50].[2] In those cases where clustering techniques are applied to corpora containing *very short* documents, further difficulties are introduced due to the low frequencies of the document terms. In this type of domains, an interesting SI algorithm named Particle Swarm Optimization (PSO) [41], has been successfully used [5, 24].

In this article, we extend a preliminary proposal of a discrete PSO algorithm named CLUDIPSO [5]. For that, we present a detailed analysis and a discussion of the results obtained with different short-text corpora. The study clearly shows that the performance of the algorithm deteriorates as the number of documents to be clustered increases. This is mainly due to the particular particle representation utilized to describe the obtained clusterings. To deal with this problem, we present a modified version of CLUDIPSO named CLUDIPSO* which incorporates modifications aimed at improving the algorithm's performance. These modifications include a new representation of particles to reduce their dimensionality, a more efficient evaluation of

[1] http://nlp.uned.es/weps/, tasks 1 and 2 on clustering information about people on the Web and clustering company tweets.

[2] In the present work, we focus on clustering methods that adequately match the manual classification criteria of human experts (or "ground truth"). This degree of correspondence is usually determined with *external validity measures* (EVM), like the entropy or the F-measure. In this context, the expressions "good performance", or "good quality" refer to those cases where the resulting groups show good values for some EVM (F-measure in our case).

the function to be optimized i.e. the Silhouette coefficient (as aftereffect of the previous modification) and some changes to the mutation operator.

The experimental work with CLUDIPSO* considers short-text corpora containing documents available on the Web (scientific abstracts, news and short legal documents) that significantly differ in number of documents, number of terms per document, number of groups and vocabulary overlapping, among others. The results are compared with those obtained by other three representative clustering algorithms: K-Means [33], K-MajorClust [25] and CHAMELEON [29].

The remainder of the paper is organized as follows. Section 2 gives a brief description of previous interesting works on clustering of short texts. Section 3 presents some general considerations about the perspective of considering clustering as an optimization problem, describing the cluster validity measure used as objective function to be optimized. Section 4 describes the previous version of the algorithm (CLUDIPSO) and Section 5 proposes the new improved version (CLUDIPSO*). In Section 6 some general features of the corpora used in the experiments are presented. The experimental setup, the analysis of the results obtained from the empirical study and the computational complexity analysis is provided in Section 7. Finally, some general conclusions are drawn and possible future works are discussed in Section 8.

2. Related Works

Clustering of (short) texts is an active research field which can be analyzed from different point of views such as efficiency, effectiveness, difficulty of the task and diversity of the approaches to address it. In this section, we first consider some efficient text and short-text clustering methods that have recently obtained good quality results. Then, the difficulties of document clustering in general and short-text clustering in particular are analyzed considering the limitations of common methods to reflect real semantics. Next, some recent approaches that attempt to overcome these limitations are described. Finally, a few approaches based on the main technique used in our proposal (PSO) are briefly explained.

The development of algorithms that produce good quality groupings efficiently is a very relevant issue in text clustering. For instance, in HSCLUST [15], a pure Harmony Search algorithm adapted for clustering tasks is combined with K-Means in three different ways: replacing the refining stage of HSCLUST with K-Means, running K-Means after each iteration of HSCLUST

and using the result in the next iteration of the algorithm, and improving the clustering obtained with HSCLUST with a one-step K-Means. These combinations allow to improve the quality of the clusters of documents in an efficient way. Another technique clustering which is simple, fast and effective is through the recursive propagation of information (messages) between documents. This idea is proposed in the *Affinity Propagation* method [19]. In that algorithm, the clusters are represented by a subset of *exemplars* (chosen randomly at first) and iteratively the method finds high quality exemplars (refining the clusters) and the corresponding clusters emerge.

Short-text clustering poses data sparseness and instability problems that directly affect the quality of the obtained results. An alternative to improve those results is by incorporating internal and external semantics to a clustering method. In [23] the authors proposed a framework with these characteristics arguing that internal semantics provides a deep understanding of texts through the use of a three-level hierarchical view while that external semantics incorporates concepts derived from multiple resources as Wikipedia and WordNet. The aspect of quality in the results of short text clustering is also studied in [42]. This study considers several clustering algorithms and different similarity measures: Cosine Similarity (CS), Latent Semantic Analysis (LSA), Short Text Vector Space Model (SVSM) and Kullback Leiber Distance (KLD). Three Hierarchical Agglomerative Clustering (HC) algorithms were analyzed: Single Link HC, Complete Link HC and Average Link HC; also the Spectral Clustering (SPEC). The results allow to conclude that the considered metrics do not always represent the correct quality of the clusters.

These difficulties that document clustering poses to current standard methods have also been analyzed from a cognitive science perspective. Text clustering methods heavily rely on a similarity measure that is supposed to adequately reflect the semantic closeness between documents. For instance, standard methods as the *vector space model* (VSM) (the method used in the present work) represent documents in a high-dimensional space, in which each dimension of the space corresponds to a word in the document collection. Here, the underlying metaphor consists in using *spatial proximity*, frequently determined by geometric measures like the cosine similarity, for *semantic proximity* [36]. This approach, the same as other simple approaches based on co-occurrences of words [48], will have serious problems to catch some subtle semantics that human beings use in speech and writing.

Cognitive sciences have provided a lot of interesting examples of why standard approaches can have trouble extracting real "semantic" information

from texts alone [16, 17, 18, 32]. French argues in [16] against the approaches that separate representations of reality from the process of manipulating them. His hypothesis is that every thing might be eventually seen as similar to another thing[3] so, he points out the necessity for context-dependent "similarity comparisons" and process-interactive representations. These ideas are extended by French and Labiouse in [17] where some failures are identified in a well-known method based on co-occurrences of words (PMI-IR) to answer rather simple subcognitive questions. These failures of PMI-IR to find human-like answers to these simple questions are attributed to its inability to understand the relational and contextual attributes of the words/concepts in the queries. Those findings are more deeply analyzed in [18] and four main problems are identified to capture human-like semantics: the intrinsic deformability of semantic space, the inability to detect co-occurrences of (especially distal) abstract structures, their lack of essential world knowledge, which humans acquire through learning or direct experience with the world and their assumption of the atomic nature of words. More recently, in [32] those criticisms are extended to a broader context and it is argued that pattern recognition, in cognitive science and related disciplines, does not accurately reflect human psychology.

The above limitations of text-only approaches are expected to get worse when short texts are analyzed. Without any contextual information and only a small number of words available in the document, achieving semantic comparisons at a level acceptable with respect to analogy-making in human beings is an even more challenging issue. Many works have focused on this aspect by proposing enriched text representations and proximity metrics that attempt to get more realistic semantic comparisons. These approaches have included the use of additional information obtained from the Web [39, 52], external resources like Wikipedia [2] and Wordnet [21], combinations of internal and external semantics [23] and learning term-weighting functions for similarity measures [51]. Although these proposals are still far from getting the semantic level previously explained in the cognitive science works, they present interesting research lines for future work.

[3]This is the idea that makes possible to find some similarities even between coffee cups and old elephants as claimed in the article's title. One might find infinite additional examples on the fluidity of language to create analogies between very distant concepts and consider, for instance, that DNA is like a staircase, DNA is like a fingerprint on a crime scene, DNA is like a zipper, and so on.

Finally, the excellent results that PSO approaches have recently obtained as global optimization methods have led many researchers to use them to solve general clustering problems. However, few adaptations have been presented for document clustering. Most are combinations of PSO with traditional clustering algorithms as K-Means [8, 40, 49]. In [8] a global search process is carried out by the PSO algorithm and then the best result obtained by the PSO algorithm is used for determining the initial centroids of the K-Means algorithm. In [40] the authors integrated the simplicity of the K-Means algorithm with the effectiveness of PSO in a unique algorithm while in [49] the result obtained by a K-Means algorithm is used as a single particle in the initial swarm of the PSO algorithm. These combinations of PSO with a clustering algorithm aim to improve the capabilities to obtain good clusters.

As summary, we can see that (short) text clustering is a challenging task that has been addressed with very different approaches which attempt to obtain good quality results with efficient algorithms. Our aim in this work is obtaining a method (CLUDIPSO*) that combines the capabilities of PSO approaches to obtain good quality results and the efficiency aspects that allow the method to scale-up to larger corpora. In this context, our proposal is a relatively simple algorithm that does not require complex document representations neither combinations with other clustering algorithms. However, as we will see in Section 8, combining both approaches is an alternative to be considered as extension of CLUDIPSO* in future works.

3. Clustering as an optimization problem

Document clustering is an automatic analytic process that assigns documents to unknown categories. In this task, only the inherent structure of data is considered; therefore, it is more difficult than supervised text categorization because no information about correctly categorized examples is provided in advance.

Clustering algorithms explore documents by searching consistent patterns among them, forming thus some groups. Many algorithms have been proposed to cluster documents [27] which can be classified into two main groups [13]: Hierarchical Clustering and Partitional Clustering.

In clustering problems, the quality of the obtained clusters cannot usually be evaluated by using typical *external* measures like F-measure or Entropy,

because the correct categorizations specified by a human expert are not available. Hence, the quality of the resulting groups is normally evaluated with respect to *structural* properties expressed in different *Internal Clustering Validity Measures* (ICVMs). These measures should ideally reflect the quality of results from a user point of view. Classical ICVMs [26, 44] 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* and the λ-measure [44].

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 phase. They can also be used as *objective functions* that the clustering algorithm attempts to optimize *during* the grouping process. This approach has been adopted by classical clustering algorithms like K-Means [33], which implements a gradient descent technique with the goal of minimizing the cluster *Sum of the Squared Error* (SSE)[4]. Cobweb [14], Autoclass [7] and CLUTO [53] also treat clustering as an optimization problem and, in these cases, the criterion function is usually explicit and can be easily stated. As observed in [53], it is possible to distinguish in this class of algorithms two key components: 1) the objective function that the clustering tries to optimize, and 2) the actual algorithm that achieves this optimization.

Regarding the first issue, it is important to observe that good values of the objective function (ICVM in this case) not always guarantee the quality of the obtained cluster from the user point of view. Therefore, a key aspect in these cases is to select as objective functions those ICVMs that have shown an adequate *correlation* degree with the categorization criteria of a human expert. In previous works on clustering of short-text corpora [10], the GS coefficient has shown very interesting results with respect to this kind of correlation. Therefore, GS was selected as objective function in the present work.

The GS measure combines two key aspects to determine the quality of a given clustering: *cohesion* and *separation*. Cohesion measures how closely related are the objects in a same cluster whereas separation quantifies how distinct (well-separated) a cluster from other clusters is. The GS coefficient of a clustering is the average cluster silhouette of all the obtained groups.

[4]SSE is a well-known measure of the quality of a clustering.

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 that is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{1}$$

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 that for this coefficient values as close to 1 as possible are desirable.

ICVMs like the GS coefficient can be used for driving clustering algorithms or for evaluating their results. However, the *real* effectiveness of these algorithms can only be evaluated with *external* measures that take into account the categorization criteria of the users. A very popular external validity measure is the F-measure [35], which combines both, *precision* and *recall*. The highest F-measure value that a grouping can obtain is 1. This value corresponds to a "perfect" categorization, i.e., a grouping that exactly matches the clustering specified by a human expert.

In the experimental work of this article, the GS coefficient was used as objective function to be optimized. The F-measure values obtained in each case were analyzed to determine the real effectiveness of the clustering algorithms. The PSO-based algorithms implemented to optimize that ICVM are described in Sections 4 and 5.

4. Particle Swarm Optimization: The CLUDIPSO algorithm

In Section 2, different clustering methods that combined PSO with traditional clustering algorithms like K-means, were described. Our PSO-based proposal instead, does not require any kind of combination with another algorithm and only relies on a standard PSO algorithm. Subsection 4.1 presents a brief description of a basic PSO algorithm and Subsection 4.2 presents a summary of CLUDIPSO, the previous version of the PSO discrete version for clustering which is extended in this work and will be described in Section 5.

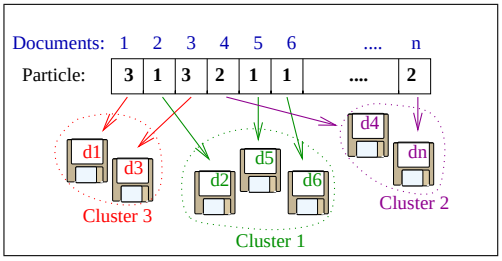


Figure 1: CLUDIPSO particle representing a clustering of n documents in 3 clusters.

4.1. The basic PSO algorithm

PSO [9] algorithm operates on a population of particles. Each particle is a real numbers vector which represents a position in the search space defined by the variables corresponding to the problem to solve. The best position currently found for the swarm (*gbest*) and the best position reached by each particle (*pbest*) are recorded at each cycle (iteration of the algorithm). The particles evolve at each cycle using two updating formulae, one for velocity (Equation (2)) and another for position (Equation (3)).

$$v_{id} = w(v_{id} + \gamma_1(pbest_{id} - par_{id}) + \gamma_2(gbest_d - par_{id})) \quad (2)$$

$$par_{id} = par_{id} + v_{id} \quad (3)$$

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 [41] whose goal is to balance global exploration and local exploitation, γ_1 is the personal learning factor, and γ_2 the social learning factor. $pbest_{id}$ is the *pbest* of the particle i at dimension d and $gbest_d$ is the *gbest* of the swarm, at dimension d.

4.2. A PSO discrete version

CLUDIPSO (CLUstering with a DIscrete Particle Swarm Optimization), is a discrete version of a PSO [9] algorithm. In CLUDIPSO [5], each valid clustering is represented as a *particle*, a n-dimensional integer vector, where n is the number of documents in the corpus. Each position in a particle corresponds to a document of the collection and the integer value stored in this position identifies the group that the document belongs to. Figure 1 illustrates a valid clustering (represented by a particle) of n documents grouped in 3 different clusters. Since the task of optimization was modeled with a discrete approach, a new formula was developed for updating the

positions (shown in Equation (4)) and the formula to update the velocities is the same than in the basic PSO (previously shown in Equation (2)). The modification in the position updating formula was introduced to accelerate the convergence velocity of the algorithm.

It is important to note that in this approach, the process of updating the position of the particles is not as direct as in the continuous case (i.e. the basic PSO was proposed to solve continuous problems). In CLUDIPSO, the positions updating process is not carried out on all dimensions at each iteration. In order to determine which dimensions of a particle will be updated, the following steps are performed: 1) all dimensions of the velocity vector are normalized in the [0, 1] range, according to the process proposed by Hu et al. [22] for a discrete PSO version; 2) a random number $r \in [0, 1]$ is calculated; 3) all the dimensions (in the velocity vector) higher than r are selected in the position vector, and updated using Equation (4).

$$par_{id} = pbest_{id} \tag{4}$$

where par_{id} is the value of the particle i at the dimension d and $pbest_{id}$ is the *pbest* of the particle i at dimension d.

A dynamic mutation operation [6] is applied if the particle is the same that its own *pbest* ($par_i = pbest_i$), in order to help avoiding convergence to a local optimum [22]. The mutation operator swaps two random dimensions of the particle and it is applied to each individual with probability pm. This value is calculated considering the total number of iterations in the algorithm (*cycles*) and the current cycle number as Equation (5) indicates:

$$pm = max\_pm - \frac{max\_pm - min\_pm}{max\_cycle} * current\_cycle \tag{5}$$

where max_pm and min_pm are the maximum and minimum values that pm can take, max_cycle is the total number of cycles that the algorithm will iterate, and $current\_cycle$ is the current cycle in the iterative process.

The GS coefficient is the objective function to be optimized by CLUDIPSO. In each iteration of the algorithm, it was used to calculate the fitness of each particle and then, the best values will be used as *gbest* and *pbest* values for the whole swarm.

5. CLUDIPSO*: An improved version of CLUDIPSO

The CLUDIPSO algorithm described in Section 4, showed competitive results in experimental studies on clustering of small corpora of short-texts

[5]. However, additional studies with larger short-text corpora (see Section 7) showed some deficiencies of CLUDIPSO as the dimensionality of corpora increased[5]. With the purpose of improving the CLUDIPSO's performance in those particular situations, a new version named CLUDIPSO* is introduced. The next subsections describe three specific characteristics that were added to the CLUDIPSO algorithm, in order to cope with larger collections: a) a new representation of particles to reduce their dimensionality and, in that way, improve the computational time of the algorithm, b) speeding up the Silhouette computation, and c) modifications to the mutation operator to obtain better clusters.

5.1. New representation and initial sub-grouping

In CLUDIPSO, the size of a particle is directly proportional to the dimensionality of the corpus to be clustered. As the number of documents in corpora increases, the updating process of CLUDIPSO becomes extremely slow. An alternative to make the particle's size manageable is to associate each position in a particle with a small group of documents X instead of a unique document as before. In that way, the integer value stored in this position identifies the group where the documents in X belong to. More formally, if p is a particle representing some arbitrary grouping $\mathcal{G} = \{G_1, \ldots, G_g\}$ and $p[i]$ is the i-th component of p, an integer k stored in $p[i]$ means that, according to the clustering $\mathcal{G}$, all the documents in the sub-group X_i belong to the group G_k (with $k \in [1, g]$). As an example, in Figure 2 (b) a CLUDIPSO*'s particle representing a particular clustering is shown. In this clustering, the documents in the sub-group X_1 (represented by $p[1]$) are associated to the group G_2 because $p[1]$ contains a 2. In a similar way, it can be seen that the documents in the sub-groups X_2 and X_5 belong to group G_1, X_3 and X_4 to G_3, and so on.

With this new representation, a previous step has to be incorporated in which closely related documents are sub-grouped in the different X_i's. This processing can be carried out by a simple k-nearest neighbors (k-nn) computation for each document[6], and obtaining in that way the required sub-

[5]From now on, the expression *dimensionality* of a corpus will be used to refer to the number of documents in a corpus. In this context, expressions such as *larger* corpora or *smaller* corpora will denote corpora with more o less documents but not necessarily imply a bigger (or smaller) size in bytes.

[6]For "k-nn" computation we mean here a basic algorithm that simply takes a document

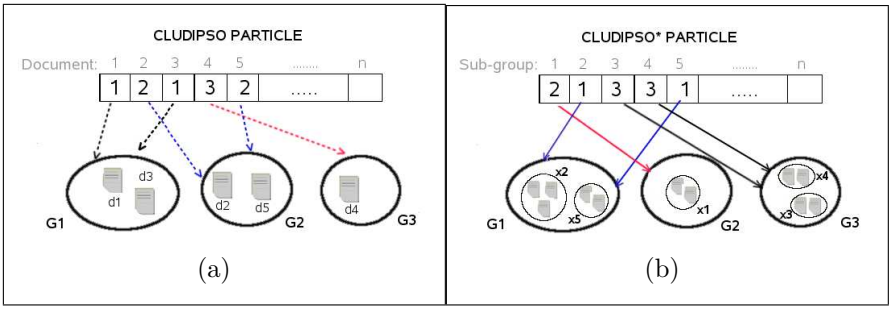


Figure 2: Particle's representation in CLUDIPSO (a) and CLUDIPSO* (b) algorithms.

groups (X_i's). As a consequence of this new representation and the initial sub-group process, a significant reduction in the dimensionality of particles was achieved[7] and accordingly in the search space.

5.2. Speeding up the Silhouette computation

A consequence of the previous modification to the CLUDIPSO algorithm, is that some information about the final classification of documents is known in advance. For instance, it is known that all the documents belonging to the same sub-group X_i will be included in the same group G_l in the final clustering ($X_i \subseteq G_l$). This information allows, for example, to pre-calculate the distances (or similarities) from a document to *all the documents in the same* X_i. In that way, it is possible to speed up the Global Silhouette computation (Section 3) when the fitness function has to be evaluated by the CLUDIPSO* algorithm. This small modification allows to obtain a significant reduction in the runtime requirements. Below, this improved process of the Silhouette computation is briefly described in two main steps: 1) T-table generation and 2) Silhouette computation based on the information contained in T. The first step is carried out *before* CLUDIPSO* is executed and the second one *during* the execution of the algorithm.

1. **T-table generation.**
   - For every document d_j, and every sub-group X_i, compute the similarity between d_j and X_i. This value, denoted $\widehat{sim}(d_j, X_i)$, is

and considers the group formed by its k nearest neighbors. It must not be confused with the classical (supervised) k-nn classification algorithm. In this study, a value $k = 3$ was empirically selected after several experiments.

[7]This reduction fluctuates between 60 and 75 percent of the original size.

computed as $\widehat{sim}(d_j, X_i) = \sum_{d_k \in X_i} sim(d_j, d_k)$, where $sim(d_j, d_k)$ is the similarity between the documents d_j and d_k.

- Each value $\widehat{sim}(d_j, X_i)$ obtained in the previous step, is stored in a $n \times d$ table (T-table) where n is the number of documents and d is the number of groups X_i. In that way, each entry $T[d_j, X_i]$ in T, will maintain the $\widehat{sim}(d_j, X_i)$ value obtained in the previous step.

2. **(Improved) Silhouette evaluation.** To evaluate the Silhouette function, the $a()$ and $b()$ sub-functions (Equation (1)) can be now more efficiently computed using the information stored in T-table:

   - Let X_k the sub-group where i was assigned to, in the initial sub-grouping process. Let G_l be the group that X_k was assigned to, in the particle p ($p[k] = l$). Let $\mathcal{X}_k = X_{k_1}, \ldots, X_{k_r}$ the sub-groups that were assigned in p the same group as X_k (G_l). It is direct to observe that the value of $a(i)$ can be directly obtained from T, by simply computing the value of $\sum_{X_{k_v} \in \mathcal{X}_k} T[i, X_{k_v}]$, and dividing this value by the number of documents in G_l ($|G_l|$).
   - A similar reasoning is required to obtain $b(i)$ from the pre-computed values in T, but considering in this case the averaged distances (or similarities) to all the groups *different* from the group that i belongs to (G_l), and selecting that averaged distance corresponding to the nearest cluster.
   - Once the required $a(i)$ and $b(i)$ values are obtained, the Silhouette coefficient $s(i)$ for the document i can be directly computed.

5.3. Modifications to the mutation operator

To help avoiding convergence to a local optimum, CLUDIPSO uses a dynamic mutation operator which is applied to each individual with probability *pm*. In CLUDIPSO* two modifications related to this operator, are introduced. Firstly, permutations are favored by increasing the probability *pm*. Then, a greater diversity in the particles exploring the search space is usually achieved. Secondly, the mutation operator swaps more than two random dimensions of the particle. A swap value close to the five percent of the particle size is used. We made a study considering several values of mutation and we empirically concluded that a value lower than that does

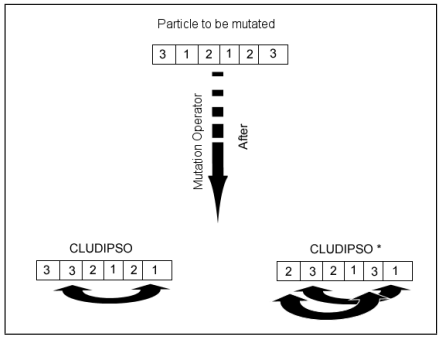


Figure 3: Modified Mutation Operator: several swaps in CLUDIPSO* versus just one swap in CLUDIPSO.

not have effect in adding diversity to the particle (changes in the structure of the solution), but a value higher than that is not recommended because the process of swapping might make the algorithm very slow. Thus, the mutation operator helps the exploration of the algorithm allowing it escape from local optimum (in the search space) which is fairly common in larger corpora. Figure 3 illustrates this modification introduced in the mutation operator of CLUDIPSO*.

6. Short-text Corpora

For the experimental work, eleven corpora with different levels of complexity with respect to the size, length of documents and vocabulary overlapping were selected: Micro4News, EasyAbstracts, SEPLN-CICLing, CICLing-2002, R4, R6, R8B, JRC6, R8-Test, JRC-Full and R8-Train. Table 1 shows some general features of these corpora: corpus size in Kbytes (CS), number of categories and documents ($|\mathcal{C}|$ and n respectively), total number of terms in the corpus ($|\mathcal{T}|$), vocabulary size ($|\mathcal{V}|$) and average number of terms per document ($\overline{\mathcal{T}}_d$). RH, which stands for *Relative Hardness*, is a specific measure which aims at estimating how related the topics corresponding to the different categories of the grouping are and, therefore, how difficult a corpus would be for a clustering task. An alternative to estimate this aspect consists in using a simple vocabulary overlapping calculus among the vocabularies of the distinct categories of the corpus. Different set overlapping measures could be used for this purpose and, in the present work, the Jaccard coefficient among the vocabularies of the categories was used. This measure, named RH in [38]

is defined as follows: let $\mathcal{C}$ a corpus with c categories $Cat_1, \ldots, Cat_c$; the *Relative Hardness* of $\mathcal{C}$, $RH(\mathcal{C})$, is defined as

$$RH(\mathcal{C}) = \frac{1}{c(c-1)/2} \times \sum_{j,k=1;j<k}^{c} sim(Cat_j, Cat_k)$$

where the similarity (sim) between two categories Cat_j, Cat_k, $sim(Cat_j, Cat_k)$, is calculated as $sim(Cat_j, Cat_k) = \frac{|Cat_j \bigcap Cat_k|}{|Cat_j \bigcup Cat_k|}$.[8]

The first eight corpora are considered small collections.[9] Micro4News, EasyAbstracts, SEPLN-CICLing and CICLing-2002 consist of news and abstracts of scientific papers, with the same number of documents (48) and categories (4). These data sets were intensively used in different works [1, 10, 24] that focused on specific characteristics of the corpora such as document lengths and its closeness respect to the topics considered in these documents. However, other characteristics such as the number of groups and number of documents per group were maintained the same for all corpora in order to obtain comparable results. The study of small corpora allows a meticulous analysis of its characteristics and a detailed understanding of the results obtained in order to compare them with those achieved with larger standard corpora. Micro4News is a low complexity collection constructed with documents that correspond to four very different topics of the popular 20Newsgroups corpus. EasyAbstracts, SEPLN-CICLing and CICLing-2002, correspond to short-length documents (abstracts of scientific papers) that mainly differ in the closeness among the topics of their categories. Thus, the EasyAbstracts corpus with scientific abstracts on well differentiated topics can be considered a medium complexity corpus but the CICLing-2002 corpus with narrow domain abstracts is a relatively high complexity corpus. This corpus, generated with abstracts of articles presented at the *CICLing 2002* conference[10] is a well-known short-text corpus that has been recognized in different works [1, 5, 10, 24, 26, 34, 37] as a very difficult corpus. The next three small corpora are subsets of the well known R8-Test corpus, a subcollection of the Reuters-21578 dataset. These corpora were artificially generated to consider corpora with different number of groups: four groups (R4), six groups (R6) and eight groups (R8B). JRC6

[8] In the sim formula each category Cat_j is considered as the "document" obtained by concatenating all the documents in Cat_j.

[9] Considering small a corpus with less than 1000 documents.

[10] http://www.cicling.org/2002/.

Table 1: Characteristics of the corpora used in the experimental work.

Corpora	CS	$\|\mathcal{C}\|$	n	$\|\mathcal{T}\|$	$\|\mathcal{V}\|$	$\overline{\mathcal{T}}_d$	RH
Micro4News	706	4	48	125614	12785	2616.95	0.16
EasyAbstracts	62	4	48	9261	2169	192.93	0.18
SEPLN-CICLing	25	4	48	3143	1169	65.48	0.14
CICLing-2002	23	4	48	3382	953	70.45	0.22
R4	184	4	266	27623	4578	166.4	0.19
R6	313	6	536	53494	4600	99.8	0.21
R8B	415	8	816	71842	5854	88.04	0.19
JRC6	9807	6	563	1424074	85605	2529.43	0.11
R8-Test	767	8	2189	150430	9315	64.87	0.07
JRC-Full	31332	6	2816	824303	109371	293.24	0.14
R8-Train	2152	8	5485	416431	15648	71.32	0.05

refers to a subcollection of JRC-Acquis [45], a popular corpus with legal documents and laws corresponding to different countries of the European Union. In order to experiment with short texts, this sub-collection only contains some of the shortest documents of six different groups of the original JRC-Acquis.

Three larger corpora were used to test the performance of the algorithms in order to study their capabilities when dealing with larger amount of documents. These corpora are cataloged as medium size.[11] The complete versions of R8-Test and R8-Train corpora were considered in this work. Also, a larger version of JRC6 corpus named JRC-Full containing a larger amount of short documents (in fact, all the short texts of six categories) was considered.[12]

For all corpora, the documents were represented using the VSM approach with the standard (normalized) tf-idf codification after a *stop-word* removal process. The popular *cosine measure* was used to estimate the similarity between two documents.

7. Experimental Results

In the experiments, 50 independent runs per corpus were performed, with 10,000 iterations (*cycles*) per run. CLUDIPSO and CLUDIPSO* used the

[11] Corpora are considered medium size if the number of documents is between 1000 and 10000.

[12] These corpora can be accessed for research purposes at: https://sites.google.com/site/lcagnina/research.

following parameters: swarm size = 50 particles, dimensions of each particle (CLUDIPSO) = number of documents (n), (CLUDIPSO*) = number of initial subgroups (approximately $n/3$), $pm\_min = 0.4$, $pm\_max = 0.9$ (CLUDIPSO), $pm\_min = 0.1$, $pm\_max = 0.9$ (CLUDIPSO*), 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 several experiments. It is important to note that for larger corpora, CLUDIPSO was tested with more iterations and more particles. It obtained with those settings the best value in the last cycles but the improvements in the performance were not substantial compared to the increment in the execution time of a single run. 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 corpus. The reported results correspond to the minimum (F_{min}), maximum (F_{max}) and average (F_{avg}) F-measure values.

The results of CLUDIPSO* were compared with those obtained by other three algorithms: K-Means [33], K-MajorClust[13] [25] and CHAMELEON [29]. K-Means is one of the most popular clustering algorithms whereas K-MajorClust and CHAMELEON are representative of the density and graph-based approaches to the clustering problem and have shown interesting results in similar problems.

7.1. CLUDIPSO* performance evaluation

CLUDIPSO* was introduced to overcome the limitations of CLUDIPSO observed with medium-size corpora (more than 1000 documents) of short texts. Analogously to the previous experiments carried out with the CLUDIPSO algorithm (see [24]), the same experimental design was reproduced with CLUDIPSO*.

Table 2 shows the F-measure values obtained by each algorithm with the smallest corpora of 48 documents. The best values are shown in bold. CLUDIPSO* achieved as good results as CLUDIPSO in Micro4News and EasyAbstracts corpora. However, the F_{min} values for SEPLN-CICLing and CICLing-2002 were slightly improved by CLUDIPSO*, with respect to those obtained

[13]The K-MajorClust algorithm is based on the MajorClust algorithm proposed in [43], but it was modified to generate exactly K groups. This modification allowed to make its results comparable to those obtained by the remaining algorithms which always generate clusterings with K groups.

Table 2: F-measure values for small corpora: Micro4News, EasyAbstracts, SEPLN-CICLing and CICLing-2002 corpora.

	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**
CLUDIPSO*	0.93	0.85	**1**	**0.92**	**0.85**	**0.98**	**0.72**	**0.68**	**0.85**	**0.62**	**0.57**	**0.73**

Table 3: F-measure values for Reuters-derived and JRC6. Small corpora with larger number of documents.

	R4			R6			R8B			JRC6		
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	0.73	0.57	0.91	0.63	0.51	0.81	0.64	**0.48**	0.78	**0.52**	0.4	**0.64**
K-MajorClust	0.74	0.45	0.79	0.53	0.36	0.74	0.5	0.28	0.68	0.44	0.33	0.55
CHAMELEON	0.61	0.47	0.83	0.52	0.42	0.66	0.57	0.30	0.71	0.38	0.31	0.56
CLUDIPSO	0.64	0.48	0.75	0.31	0.26	0.38	0.21	0.18	0.28	0.30	0.26	0.33
CLUDIPSO*	**0.87**	**0.81**	**0.93**	**0.71**	**0.61**	**0.83**	**0.72**	0.47	**0.80**	0.48	**0.42**	0.57

by CLUDIPSO. A similar situation was observed with the F_{avg} value for CICLing-2002, which was slightly improved with the new version of the PSO algorithm.

Table 3 shows F-measure values obtained by each algorithm with the four small corpora with largest number of documents. CLUDIPSO* reached the best F_{avg} and F_{max} values with R4, R6 and R8B corpora with a considerable difference with respect to the other tested algorithms. The same happened with F_{min} metric for those corpora except for R8B for which K-Means reached the best value. This fact could also be observed examining the corresponding boxplot in Figure 4. With respect to JRC6 corpus, CLUDIPSO* outperforms the results obtained by all algorithms, except K-Means which reached slightly better F_{avg} and F_{max} values. If we compare the values obtained with CLUDIPSO and CLUDIPSO* in Table 3, it can be seen that CLUDIPSO* clearly outperformed CLUDIPSO in these corpora, with improvements of performance oscillating between 24% and 212%.

Table 4 shows F-measure values for the medium size corpora R8-Test, JRC-Full and R8-Train. Observe that CLUDIPSO's results are not shown because its performance is very poor hence not comparable with the rest of the algorithms. The F_{avg}, F_{min} and F_{max} values obtained with CLUDIPSO* are similar or better than those of the rest of the algorithms. For R8-Test and JRC-Full corpora, K-Means reached F_{max} values similar to CLUDIPSO* although

Table 4: F-measure values for medium size corpora: R8-Test, JRC-Full and R8-Train.

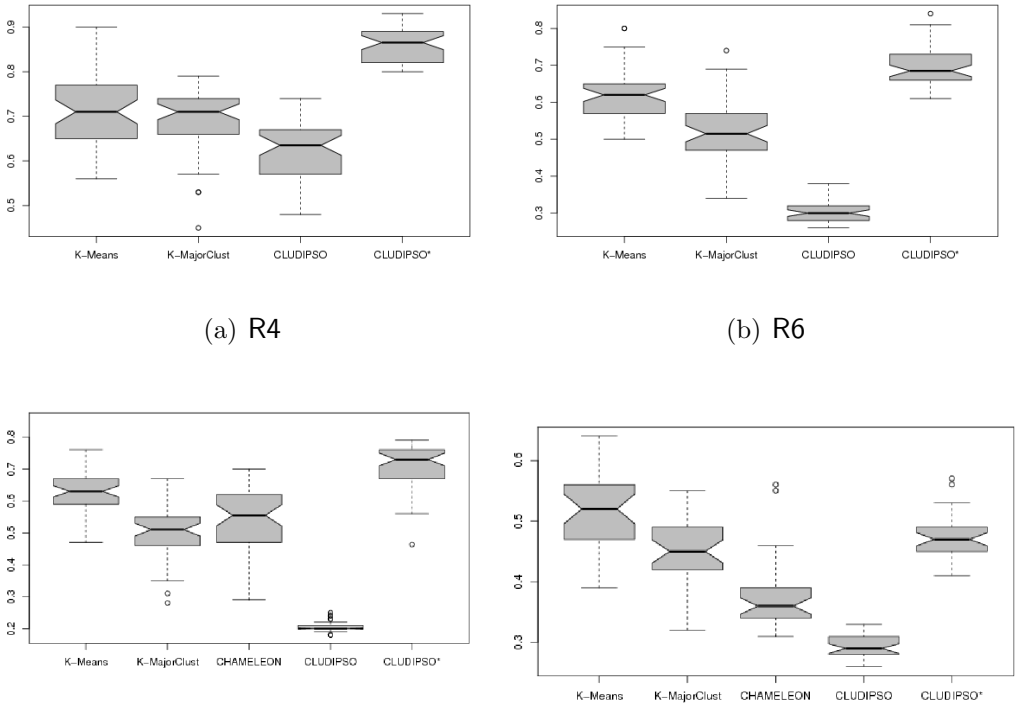
	R8-Test			JRC-Full			R8-Train		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means	0.67	0.54	**0.71**	0.50	0.44	**0.56**	0.60	0.45	0.74
K-MajorClust	0.53	0.44	0.62	0.47	**0.46**	0.50	0.55	0.45	0.59
CHAMELEON	0.56	0.53	0.59	0.47	0.38	0.47	NA	NA	NA
CLUDIPSO*	**0.70**	**0.69**	**0.71**	**0.51**	**0.46**	**0.56**	**0.78**	**0.76**	**0.81**

the latter obtained F-measure values with the smallest dispersion [0.69,0.71] and [0.46,0.56] respectively. For these latter two corpora CLUDIPSO* outperforms the results of K-MajorClust and CHAMELEON. With respect to the results for R8-Train corpus, CLUDIPSO* outperforms K-Means and K-MajorClust in 30% and 45% of the results respectively (on average). For this corpus, CHAMELEON was not able to obtain any result.[14]

A statistical analysis of variance over the F-measure values complements our performance study. This analysis can be obtained from the boxplots with the distribution of (averaged) F-measure values of each algorithm with all the corpora.[15] Figure 4 shows the F-measure values distribution obtained by the algorithms only with the small Reuters derived-corpora and JRC6 corpus, because with the corpora with the smallest number of documents (48), the boxplots are similar (the same results were obtained) to those of CLUDIPSO [24]. The boxplots for R4 in Figure 4 (a) show that CLUDIPSO* reached the best performance for this corpus, with a median value close to 0.9 and a little bias to the right side. K-Means and K-MajorClust have considerable dispersion and close median values. The boxplot corresponding to CLUDIPSO shows the poor performance obtained with this corpus. With the R6 corpus (Figure 4 (b)) something similar occurs and the distribution of F-measure values obtained with CLUDIPSO* and K-Means are similar. The boxplot of CLUDIPSO shows the lowest quality in the obtained results. In Figure 4 (c) it is possible to observe that with R8B, CLUDIPSO* obtained the best values. The worst values were obtained with CLUDIPSO although with a

[14]The algorithm was executed in an Intel(R) Core(TM)2 Quad CPU 2.83GHz 3GB RAM and it failed because of the lack of RAM memory to complete the process.

[15]Boxplots [47] are descriptive statistical tools for displaying information (dispersion, quartiles, median, etc.) among populations of numerical data, without any assumptions about the underlying statistical distribution of the data. CHAMELEON is not considered in some boxplots because this algorithm obtains, for some corpora, an insufficient number of results to make the distributions comparable from a statistical point of view.



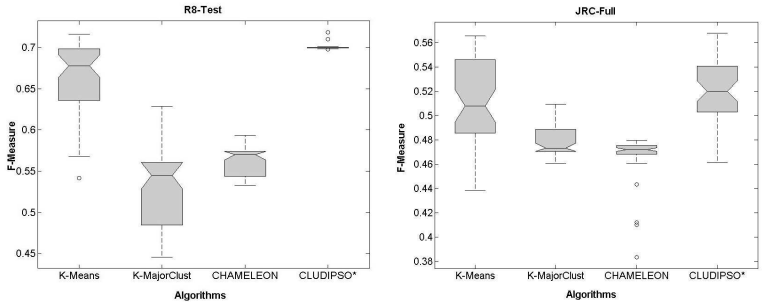
(a) R4 (b) R6

(c) R8B (d) JRC6

Figure 4: Boxplots for JRC6 and Reuters-derived corpora (CLUDIPSO⋆).

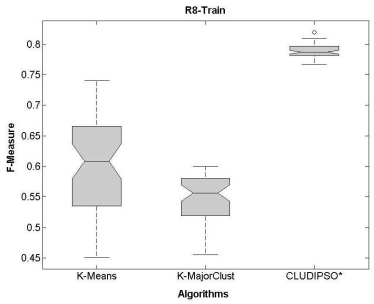
minimum dispersion (flat box) indicating similar values of F-measure in most of the runs. CHAMELEON presents the highest dispersion of values but its results are better than K-MajorClust and CLUDIPSO ones. With respect to the JRC6 corpus (Figure 4 (d)), CLUDIPSO⋆ and CLUDIPSO evidence the lowest dispersion, compared with the other algorithms. The median value of CLUDIPSO⋆ is between the median values of K-MajorClust and K-Means, being the last one the best obtained value. The evident differences observed in the boxplots show a significant complexity of the corpus and a wide variation in the performance of the considered algorithms.

The boxplots for R8-Test corpus in Figure 5 (a) show that CLUDIPSO⋆ reached the best performance with a median value close to 0.7 and without dispersion except for some outliers. This aspect is important because it



(a) R8-Test

(b) JRC-Full



(c) R8-Train

Figure 5: Boxplots for R8-Test, JRC-Full and R8-Train.

shows that the algorithm was able to find very similar F-measure values in all the runs. The worst performance corresponds to K-MajorClust because its median value is the lowest and its boxplot shows the highest dispersion. Figure 5 (b) shows the boxplots for JRC-Full corpus. It is possible to observe that the best median value was obtained with CLUDIPSO* although the one obtained with K-Means is very close. The main difference between both algorithms is the dispersion observed for each one, being the highest that of K-Means. CHAMELEON shows the minimum dispersion of values. The boxplots for R8-Train corpus are shown in Figure 5 (c). It is worth noting that CLUDIPSO* not only obtained the best F-measure value but the best median value and the best dispersion (the lowest compared with those of the rest of the algorithms).

As final conclusion of this study, it is possible to state that CLUDIPSO* obtained good results for all the tested corpora. In particular, a simple comparison of the boxplots of CLUDIPSO and CLUDIPSO* (Figure 4) shows a

clear improvement in the results obtained by the latter. CLUDIPSO* outperforms (in few cases obtained similar values) K-MajorClust and CHAMELEON in all corpora. The same happened if our approach is compared with K-Means except for R8B and JRC6 corpora. In those, it is not possible to state which algorithm (K-Means or CLUDIPSO*) has the best performance (see Table 3 with the best F-measure values). K-Means obtained the best F_{avg} and F_{max} values but the best F_{min} value was obtained with CLUDIPSO*. For a deeper investigation of the behavior of both algorithms (K-Means and CLUDIPSO*) for those corpora (R8B and JRC6), the Kruskal-Wallis [20] and Tukey [47] tests were calculated. The Kruskal-Wallis nonparametric one-way analysis was applied because the values (the samples) do not have a Normal distribution (determined with the Kolmogorov-Smirnov test [12]). The Kruskal-Wallis test returns the *p-value* for the null hypothesis for all samples (*there are a significant difference between the samples*). A *p-value* near to zero suggests that at least one sample is significantly different (or *statistically significant*) than the other samples. Usually, if the *p-value* is less than 0.05, the results are declared statistically significant. The Tukey test was used to determine under which experimental conditions the differences are significant, that is, between which pairs of data the difference is significant. The test returns a range. If the range does not contain the zero-value, then the results are confirmed (significantly different).

Table 5 illustrates the *p-value* and Tukey intervals. For the R8B corpus, there is not a statistically significant difference (*p-value*=0.461 >0.05). The Tukey interval confirms that CLUDIPSO* and K-Means are not significantly different with the R8B corpus (the range includes the zero-value). For the JRC6 corpus, there is a significant difference because of the *p-value*=0.002<0.05. The assertion is confirmed with the Tukey test because the corresponding interval does not include the zero-value. Finally, observing Table 3, it is possible to conclude that there is a statistical difference and K-Means outperformed CLUDIPSO* with the JRC6 corpus but the behavior of both algorithms was equivalent with the R8B corpus.

In summary, the performance study of CLUDIPSO* confirmed that it performs as well as CLUDIPSO with small short-text corpora and, with respect to medium-size corpora, it is possible to state that it represents a high competitive approach that outperforms state-of-the-art representative algorithms.

Table 5: Statistical comparison between CLUDIPSO* and K-Means with R8B and JRC6 corpora.

	R8B		JRC6	
Alg.	*p-value*	TukeyI	*p-value*	TukeyI
CLUDIPSO* vs. K-Means	0.461	[-0.031,0.012]	0.002	[0.013,0.073]

7.2. Computational Complexity of CLUDIPSO and CLUDIPSO*

The computational complexity of each step in both algorithms is analyzed assuming P particles, D dimensions for each particle (equivalent to the number of documents to cluster), N iterations, pm probability of mutation, G number of groups obtained with K-nearest neighbors algorithm (approximately $D/3$) and D_G the number of documents of each group G (approximately 3).

7.2.1. CLUDIPSO

The main steps involved in the process carried out by CLUDIPSO algorithm are detailed as follows.

- Step 1: the initialization of particles and velocities takes $P * D$. For the evaluation of each particle (to set the *pbest* values) is necessary to obtain the Silhouette coefficients of each particle. Silhouette coefficient computation is $\mathcal{O}(D^2)$ [3] then the evaluation of each particle takes approximately $P * D^2$. Thus, the time of step 1 uses $T_{step1} : P * D + P * D^2$.

- Step 2: the evolution of the swarm is executed using the updating equations. Although the updating of particles is not made in all dimensions (see Section 4.2), as upper bound we assume that all dimensions will be updated. Thus, this step takes $T_{step2} : P * D$.

- Step 3: the use of the mutation operator depends on the pm value which is updated in every iteration of the algorithm. For the sake of simplicity, we consider a fixed value (as worst case, $pm = 1$). Then, considering that the operator is applied to all particles, the time of step 3 uses $T_{step3} : P$.

- Step 4: the evaluation of all particles and the updating of each *pbest* value takes $T_{step4} : P * D^2 + P$.

Finally, the total computational complexity considering all the iterations is $T_{cludipso} : N * (T_{step1} + T_{step2} + T_{step3} + T_{step4}) = 2 * P * N * (D + D^2 + 1)$ which is $\mathcal{O}(D^2)$.

7.2.2. CLUDIPSO⋆

The main steps involved in the process carried out by CLUDIPSO⋆ algorithm are detailed as follows.

- Step 1: the initialization of particles and velocities takes $P * G$. The G value is obtained previously executing the K-nearest neighbors algorithm with a linear cost in the number of documents [35], i.e. $\mathcal{O}(D)$. Then, the T-table generation corresponding to each group (see Section 5.2) takes $G * D_G^2$. To obtain the fitness of each particle (to set the *pbest* values) the improved Silhouette evaluation is almost direct (see Section 5.2) and takes $P * G$. Thus, the time of step 1 uses $T_{step1} : D + P * G + G * D_G^2 + P * G$.

- Step 2: the evolution of the swarm is similar to Step 2 of CLUDIPSO (previously described) but considering G instead of D. Thus, this step takes $T_{step2} : P * G$.

- Step 3: the use of the mutation operator is similar to Step 3 of CLUDIPSO. Observe that the swap of the 5% of dimensions is not a relevant value for the complexity computation. Thus, the time of this step uses $T_{step3} : P$.

- Step 4: the evaluation of all particles and the updating of each *pbest* value takes $T_{step4} : P * G + P$.

Finally, the total computational complexity considering all the iterations is $T_{cludipso*} : N*(T_{step1}+T_{step2}+T_{step3}+T_{step4}) = 2*P*N+D*N+4*P*G*N+G*D_G^2*N$. Replacing G for $\frac{D}{3}$ we have: $2*P*N+D*N*(\frac{4*P}{3}+\frac{D_G^2}{3}+1)$ which is $\mathcal{O}(D_G^2)$. As D_G^2 value is much less than D^2 ($D_G^2 << D^2$) we conclude that the computational complexity of CLUDIPSO⋆ is smaller than the computational complexity of CLUDIPSO.

The reduction in the number of dimensions of particles in CLUDIPSO* also favors the reduction on the computational complexity of this new version. This improvement has a direct beneficial effect on the execution times of CLUDIPSO* which, although are still high for the largest corpora, are conclusively better than the CLUDIPSO's ones, mainly in corpora which have over 800 documents.

The computational complexity of K-Means is lineal in the amount of documents, that is $\mathcal{O}(D)$ [35] and $\mathcal{O}(D^2)$ for CHAMELEON [29]. K-MajorClust is based on MajorClust algorithm but uses exactly K clusters. As the computational complexity of MajorClust [28] does not depend on the number of the groups, we conclude that the latter two have similar computational complexity, that is, $\mathcal{O}(D^2)$. Comparing all computational complexities, we conclude that CLUDIPSO* has a lower value ($\mathcal{O}(D_G^2)$) than that of K-Means ($\mathcal{O}(D)$) because ${D_G}^2$ is the number of documents of each group G which is lower than the number total of document to cluster D. Also the computational complexity of CLUDIPSO* is lower than those of K-MajorClust and CHAMELEON as we explained before ($D_G^2 << D^2$).

8. Conclusions and Future Work

This work analyzed if Particle Swarm Optimization can be an effective approach to cluster short texts on the Web, such as short news and scientific abstracts. This study considered short-text corpora of small and medium dimensionality of scientific abstracts, news and small legal documents. Our research, first considered a simple discrete PSO approach, CLUDIPSO, which obtained very hight quality results with small short-text corpora with less than 50 documents. However, with larger corpora a constant deterioration in the F- measure values was observed, as the number of documents increased. In these cases, the CLUDIPSO's performance was not comparable to the performance of other traditional algorithms like, for instance, K-Means, showing evident limitations to explore these bigger search spaces. The experimental work showed that with small corpora (small search space), CLUDIPSO quickly evolved all particles in each iteration obtaining rapidly good results. However, when the search space grew (i.e., the number of documents increased), it could not converge towards good quality results.

The main contribution in this paper, was the presentation and analysis of CLUDIPSO*, an improved version of CLUDIPSO which aims at dealing with those problems. This proposal included a new particle's representation, a more efficient fitness evaluation (Silhouette computation) and a more effective mutation operator. All these modifications, allowed to reduce the search space and the time required by each cycle of the algorithm. CLUDIPSO* was compared with other traditional clustering algorithms (K-means, K-MajorClust and CHAMELEON) and obtained a good performance for all

the considered corpora, showing to be a highly competitive algorithm to cluster short-text corpora of small and medium size.

PSO approaches, as other meta-search clustering algorithms such as simulated annealing and genetic algorithms, are very appealing methods due to the great flexibility they exhibit, as optimization methods of a global goal criterion. However, their runtime requirements are typically unacceptably high. In this context, we consider that our proposal does not only show interesting results with respect to quality criteria. It also addresses some crucial problems in this kind of methods, like those related to efficiency aspects. Moreover, many of the ideas introduced in CLUDIPSO* to improve these aspects, could be easily adapted to other meta-search clustering algorithms without much effort.

As future work, we plan to implement the previous idea, and adapt the mechanisms introduced in CLUDIPSO* to other meta-search methods. We also intend to work with larger corpora (with more than ten thousands of documents) in order to determine what is the (approximated) maximum size (in number of documents) that our approach can manage without starting to show an evident deterioration in quality of results or runtime requirements of the algorithm. The study of alternative methods to perform the initial sub-grouping and start the search process with higher quality initial subgroups could be an interesting extension. Another possible extension, is to use the results obtained with CLUDIPSO* as input to a recent (and effective) boosting method [11] and obtain thus, higher quality solutions.

The representation of documents and proximity measure used in this work (VSM and cosine similarity respectively) attempt capturing the semantic similarity between two documents. On the other hand, we saw in Section 2 that much work needs to be done to obtain improved document representations and proximity measures that more adequately represent real semantics at the human level. We consider that CLUDIPSO*, which obtained very interesting results with this basic text comparison method, could be significantly improved by combining it with those "more semantic" approaches and that will be other of the aspects addressed as future work.

CLUDIPSO*, the same as the remaining algorithms analyzed in this article, requires as input parameter the number of clusters to be used. This information, could not be available for many practical problems so, as future work, we plan to address this weakness and obtain a version that automatically determines that value. A (simple) alternative is introducing a pre-processing stage that, based on intrinsic information of the generated

groups,[16] automatically estimates an adequate number of groups. However, other more elaborated (and interesting) approaches could incrementally modify the number of groups simultaneously with the operation of the clustering process.

Finally, it is clear that due to the strong tendency of people to use reduced language in text messages and recently in the framework of the so-called Web 2.0 in weblogs, social networks, or microblogs such as Twitter, the short-text clustering task is becoming even more important because of the need for analyzing comments and opinions from users. The evaluation of CLUDIPSO* in this kind of problems is a very important aspect to be considered in our future work.

9. Acknowledgements

The research work is partially funded by the European Commission as part of the WIQ-EI IRSES research project (grant no. 269180) within the FP 7 Marie Curie People Framework and it has been developed in the framework of the Microcluster VLC/Campus (International Campus of Excellence) on Multimodal Intelligent Systems. The research work of the first author is partially funded by the program PAID-02-10 2257 (Universitat Politècnica de València) and CONICET (Argentina).

References

[1] M. Alexandrov, A. Gelbukh, P. Rosso, An approach to clustering abstracts, in: A. Montoyo, R. Muñoz, E. Métais (Eds.), Natural Language Processing and Information Systems, volume 3513 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2005, pp. 1–10.

[2] S. Banerjee, K. Ramanathan, A. Gupta, Clustering short texts using wikipedia, in: Proc. of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR-2007), ACM, New York, NY, USA, 2007, pp. 787–788.

[16] See for instance [46], Chap. 8, Section 8.5.5, for an explanation of how unsupervised cluster evaluation measures can be used to approximately determine the correct or natural number of clusters.

[3] P. Berkhin, Survey Of Clustering Data Mining Techniques, Technical Report, Accrue Software, 2002.

[4] W. Bin, S. Zhongzhi, A clustering algorithm based on swarm intelligence, in: Proc. of the Int. Conf. on Info-tech and Info-net, ICII 2001, volume 3, IEEE, 2001, pp. 58–66.

[5] L. Cagnina, M. Errecalde, D. Ingaramo, P. Rosso, A Discrete Particle Swarm Optimizer for Clustering Short-text Corpora, in: Proc. International Conference on Bioinspired Optimization Methods and their Applications, BIOMA 2008, Jozef Stefan Institute, 2008, pp. 93–103.

[6] L. Cagnina, S. Esquivel, R. Gallard, Particle swarm optimization for sequencing problems: a case study., Congress on Evolutionary Computation, IEEE Press, Portland, Oregon, USA, 2004, pp. 536–541.

[7] P. Cheeseman, J. Stutz, Bayessian classification (autoclass): Theory and results, in: P.S. U. Fayyad, G. Piatetsky-Shapiro, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, American Association for Artificial Intelligence Menlo Park, CA, USA, 1996, pp. 153–180.

[8] X. Cui, T. Potok, P. Palathingal, Document clustering using particle swarm optimization, in: Proc. of IEEE Swarm Intelligence Symposium, SIS 2005, IEEE Press, 2005.

[9] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proc. of the Sixth International Symposium on Micro Machine and Human Science, MHS'95, IEEE Press, Nagoya, Japan, 1995, pp. 39–43.

[10] M. Errecalde, D. Ingaramo, P. Rosso, Proximity estimation and hardness of short-text corpora, in: Proc. of 5th Int. Workshop on Text-based Information Retrieval, TIR 2008, IEEE CS, 2008, pp. 15–19.

[11] M. Errecalde, D. Ingaramo, P. Rosso, Itsa*: an effective iterative method for short-text clustering tasks, in: Proc. of the 23rd Int. Conf. on Industrial Engineering and other Applications of Applied Intelligent Systems, IEA/AIE 2010, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 550–559.

[12] G. Fasano, A. Franceschini, A multidimensional version of the kolmogorov-smirnov test, Monthly Notices of the Royal Astronomical Society 225 (1987) 155–170.

[13] A. Feng, Document clustering - an optimization problem, in: ACM Conf. on Research and Development in Information Retrieval, ACM, 2007, pp. 819–820.

[14] D. Fisher, Knowledge acquisition via incremental conceptual clustering, Machine Learning 2 (1987) 139–172.

[15] R. Forsati, M. Mahdavi, M. Shamsfard, M. Reza Meybodi, Efficient stochastic algorithms for document clustering, Information Sciences 220 (2013) 269–291.

[16] R.M. French, When coffee cups are like old elephants or why representation modules don't make sense, in: A. Riegler, M. Peschl (Eds.), Proc. of the 1997 International Conference on New Trends in Cognitive Science, Austrian Society for Cognitive Science, 1997, pp. 158–163.

[17] R.M. French, C. Labiouse, Why co-occurrence information alone is not sufficient to answer subcognitive questions, Journal of Experimental & Theoretical Artificial Intelligence 13 (2001) 421–429.

[18] R.M. French, C. Labiouse, Four problems with extracting human semantics from large text corpora, in: Proc. of the 24th Annual Conference of the Cognitive Science Society, George Mason University, 2002, pp. 316–321.

[19] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (2007) 972–976.

[20] M. Hollander, D.A. Wolfe, Nonparametric Statistical Methods, Wiley, 1973.

[21] A. Hotho, S. Staab, G. Stumme, Ontologies improve text document clustering, in: Proc. of the Third IEEE International Conference on Data Mining (ICDM-2003), IEEE Computer Society, Washington, DC, USA, 2003, pp. 541–544.

[22] X. Hu, R. Eberhart, Y. Shi, Swarm intelligence for permutation optimization: a case study on n-queens problem, in: Proc. of the IEEE Swarm Intelligence Symposium, IEEE Press, 2003, pp. 243–246.

[23] X. Hu, N. Sun, C. Zhang, T. Chua, Exploiting internal and external semantics for the clustering of short texts using world knowledge, in: Proc. of the 18th ACM Conf. on Information and knowledge management, ACM, New York, NY, USA, 2009, pp. 919–928.

[24] D. Ingaramo, M. Errecalde, L. Cagnina, P. Rosso, Computational intelligence and bioengineering, Computational Intelligence and Bioengineering, F. Masulli and A. Micheli and A. Sperduti Eds. IOS Press, 2009, pp. 3–19.

[25] D. Ingaramo, M. Errecalde, P. Rosso, A general bio-inspired method to improve the short-text clustering task, in: Proc. of 11th Int. Conf. on Intelligent Text Processing and Computational Linguistics, CICLing 2010, volume 6008 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2010, pp. 661–672.

[26] D. Ingaramo, D. Pinto, P. Rosso, M. Errecalde, Evaluation of internal validity measures in short-text corpora, in: Proc. of the Int. Conf. on Intelligent Text Processing and Computational Linguistics, CICLing 2008, volume 4919 of *Lecture Notes in Computer Science*, Springer-Verlag, 2008, pp. 555–567.

[27] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.

[28] T. Karthikeyan, S.J. Peter, S. Chidambaranathan, Hybrid algorithm for noise-free high density clusters with self-detection of best number of clusters, International Journal of Hybrid Information Technology 4 (2011) 39–54.

[29] G. Karypis, E. Han, V. Kumar, Chameleon: Hierarchical clustering using dynamic modeling, Computer 32 (1999) 68–75.

[30] T. Krink, S. Paterlini, Differential evolution and particle swarm optimisation in partitional clustering, Computational Statistics and Data Analysis 50 (2006) 1220–1247.

[31] N. Labroche, N. Monmarché, G. Venturini, Antclust: Ant clustering and web usage mining, in: Genetic and Evolutionary Computation GECCO 2003, volume 2723 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2003, pp. 25–36.

[32] A. Linhares, D.M. Chada, What is the nature of the mind's pattern-recognition process?, New Ideas in Psychology 31 (2013) 108 – 121.

[33] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 1967, pp. 281–297.

[34] P. Makagonov, M. Alexandrov, A. Gelbukh, Clustering abstracts instead of full texts, in: Proc. of Int. Conf. on Text, Speech and Dialogue, TSD 2004, volume 3206 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 2004, pp. 129–135.

[35] C.D. Manning, P. Raghavan, H. Schutze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[36] C.D. Manning, H. Schütze, Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge, MA, USA, 1999.

[37] D. Pinto, J.M. Benedí, P. Rosso, Clustering narrow-domain short texts by using the Kullback-Leibler distance, in: Proc. of the Int. Conf. on Intelligent Text Processing and Computational Linguistics, CICLing 2007, volume 4394 of *Lecture Notes in Computer Science*, Springer-Verlag, 2007, pp. 611–622.

[38] D. Pinto, P. Rosso, On the relative hardness of clustering corpora, in: Proc. of Int. Conf. on Text, Speech and Dialogue, TSD 2007, volume 4629 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 2007, pp. 155–161.

[39] M. Sahami, T.D. Heilman, A web-based kernel function for measuring the similarity of short text snippets, in: Proc. of the 15th international conference on World Wide Web (WWW-2006), ACM, New York, NY, USA, 2006, pp. 377–386.

[40] F. Samadzadegan, S. Saeedi, Clustering of lidar data using particle swarm optimization algorithm in urban area, in: Laserscanning 09, Volume XXXVIII (2009), pp. 334–339.

[41] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Proc. of the IEEE Int. Conf. on Evolutionary Computation, IEEE Press, 1998, pp. 69–73.

[42] P. Shrestha, C. Jacquin, B. Daille, Clustering short text and its evaluation, in: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing, volume 7182 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2012, pp. 169–180.

[43] B. Stein, S. Meyer zu Eissen, Document Categorization with MajorClust, in: Proc. Workshop on Information Technologies and Systems, WITS 2002, Technical University of Barcelona, 2002, pp. 91–96.

[44] B. Stein, S. Meyer zu Eissen, F. Wissbrock, On cluster validity and the information need of users, in: Proc. of the Int. Association of Science and Technology for Development, IASTED 2003, ACTA Press, 2003, pp. 216–221.

[45] R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufis, D. Varga, The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages, in: Proc. of the 5th Int. Conf. on Language Resources and Evaluation, LREC 2006, European Language Resources Association, 2006, pp. 2142–2147.

[46] P. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, 2006.

[47] J.W. Tukey, Exploratory data analysis, Addison-Wesley Publishing Company, Reading, MA, 1977.

[48] P.D. Turney, Mining the web for synonyms: Pmi-ir versus lsa on toefl, in: Proc. of the 12th European Conference on Machine Learning, EMCL '01, Springer-Verlag, London, UK, UK, 2001, pp. 491–502.

[49] D.W. van der Merwe, A.P. Engelbrecht, Data clustering using particle swarm optimization, in: Proc. of the Congress on Evolutionary Computation, Special Session on Design Optimisation with Evolutionary Computation, CEC 2003), IEEE Press, 2003, pp. 215–220.

[50] X. Xiao, E. Dow, R. Eberhart, Z.B. Miled, R. Oppelt, Gene clustering using self-organizing maps and particle swarm optimization, in: Proc. of the 17th Int. Symposium on Parallel and Distributed Processing, IEEE Computer Society, 2003.

[51] W.T. Yih, Learning term-weighting functions for similarity measures, in: Proc. of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP '09, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 793–802.

[52] W.T. Yih, C. Meek, Improving similarity measures for short segments of text, in: Proc. of the 22nd national conference on Artificial intelligence (AAAI-2007), AAAI Press, 2007, pp. 1489–1494.

[53] Y. Zhao, G. Karypis, Empirical and theoretical comparison of selected criterion functions for document clustering, Machine Learning 55 (2004) 311–331.