**Invited Paper:**

# A new AntTree-based algorithm for clustering short-text corpora

**Marcelo Luis Errecalde, Diego Alejandro Ingaramo**
**Development and Research Laboratory in Computacional Intelligence (LIDIC)**
**Universidad Nacional de San Luis**
**San Luis, Argentina**
**{merreca,daingara}@unsl.edu.ar**

**Paolo Rosso**
**Natural Language Engineering Lab.,ELiRF,**
**Departamento de Sistemas Informáticos y Computación**
**Universidad Politécnica de Valencia**
**Valencia, Spain**
**prosso@dsic.upv.es**

## Abstract

Research work on "short-text clustering" is a very important research area due to the current tendency for people to use 'small-language', e.g. blogs, text-messaging and others. In some recent works, new bio-inspired clustering algorithms have been proposed to deal with this difficult problem and novel uses of Internal Clustering Validity Measures have also been presented. In this work, a new AntTree-based approach is proposed for this task. It integrates information on the *Silhouette Coefficient* and the concept of *attraction* of a cluster in different stages of the clustering process. The proposal achieves results comparable to the best reported results in this area, showing an interesting stability in the quality of the results and presenting some interesting capabilities as a general improvement method for arbitrary clustering approaches.
**Keywords:** Short-text clustering, Bio-inspired algorithms, AntTree, Internal Validity Measures, Silhouette Coefficient.

## 1 INTRODUCTION

Automatic document clustering is one of the most important approaches to deal with the information overload problem caused by the proliferation of documents available on the Web, corporate intranets, news wires, etc. In a nutshell, document clustering is an unsupervised process that assigns documents to unknown categories (or groups) called *clusters*, whose members are similar in some way.

Many of the most interesting potential applications of document clustering, involve "*short texts*". The Web provides us a considerable number of examples of different types of short documents that are available for automatic analysis such as emails, news, scientific abstracts, blogs, snippets, chats, FAQs and online evaluations of commercial products. In all these cases, clustering methods can play an important role to analyze and organize this huge number of short documents.

Short-text document clustering is considered a very difficult problem due to the low frequencies of the terms in the documents. However, some recent research works have started studying different aspects related to this problem. These works include the study of the correlation between internal and external validity measures [8], the estimation of the hardness of short-text corpora [11, 5] and the use of bio-inspired clustering methods [3, 7]. In all these cases, the use of Internal Clustering Validity Measures have played an important role.

A question that arises from these works is if some Internal Clustering Validity Measures, could also be used in other existing bio-inspired clustering methods, in order to improve their performance. This work addresses this aspect by proposing a new AntTree-based algorithm named *AntSA* which integrates into an unified approach the *Silhouette Coefficient* [12] and the concept of *attraction* of a cluster. AntSA is based on the AntTree algorithm [2] but it incorporates information about both measures in different stages of the clustering process.

The remainder of the paper is organized as follows. Section 2 presents some general considerations about different uses of Internal Clustering Validity Measures in short-text clustering tasks. Section 3 describes the AntSA method proposed in this work. The experimental setup and the analysis of the results obtained from our empirical study is provided in Section 4. Finally, some general conclusions are drawn and possible future work is discussed.

## 2 INTERNAL VALIDITY MEASURES AND CLUSTERING TASKS

Document clustering is the unsupervised assignment of documents to unknown categories. This task is more difficult than supervised document categorization because 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 or 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 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) (denoted $\bar{\rho}$) and the $\lambda$-Measure (see [8] and [13] for more detailed descriptions of these ICVMs).

Most of people working on clustering problems are familiar with the use of ICVMs as cluster validation tools. However, some recent works have proposed other uses of this kind of measures, specially in the context of short-text clustering problems. In [8] for example, an analysis of the correlation between distinct ICVMs and the well known $F$-Measure is presented. The evaluation of several ICVMs on the "gold standard" of different short-text collections is proposed in [5] as a method to estimate the hardness of those corpora.

ICVMs have also been used as explicit *objective functions* that the clustering algorithms attempt to optimize [6, 15]) This idea has recently been used in short-texts clustering tasks, taking the GS and the EDM $\bar{\rho}$ measures as objective functions and using discrete and continuous Particle Swarm Optimization (PSO) algorithms as function optimizers [3, 7]. In these works, a discrete PSO algorithm named CLUDIPSO obtained the best results on different short-text corpora when the GS measure was used as objective function.

The GS measure is an interesting ICVM that 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 other clusters 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 to determine the silhouette coefficient of any arbitrary object $i$, that we will refer as $s(i)$ and that is defined as $s(i) = \frac{b(i)-a(i)}{max(a(i),b(i))}$ 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 that we want for this coefficient values as close to 1 as possible.

Tacking into account that ICVMs like GS or the EDM $\bar{\rho}$ have played an important role in cluster validation, hardness estimation and optimization-based approaches to clustering, it would be interesting to investigate if these and other ICVMs could be used in other stages and processes of the clustering algorithms in order to improve their performances. In this work, a new AntTree-based algorithm named *AntSA* aims to answer this question by integrating into an unified approach the *Silhouette Coefficient* [12] and the concept of *attraction* of a cluster.

## 3 THE AntSA ALGORITHM

The *AntSA* (**Ant**Tree-**S**ilhouette-**A**ttraction) algorithm is based on the AntTree algorithm [2] but it also incorporates information related to the Silhouette Coefficient and the concept of *attraction* of a cluster in different stages of the clustering process. To understand how AntSA works, some preliminary concepts on the AntTree algorithm are neccesary. Therefore, the main ideas on the AntTree algorithm will be first introduced in subsection 3.1 before the description of the AntSA algorithm in subsection 3.2.

### 3.1 The AntTree algorithm

The AntTree algorithm [2] is based on the self-assembly behavior observed in certain species of ants where the living structures are used as bridges or auxiliary structures to build the nest. The structure is built by using an incremental process in which ants joint a fixed support or another ant for assembling. AntTree builds a tree structure representing a hierarchical data organization which divides the whole data set. Each ant represents a single datum from the data set and it moves in the structure according to its similarity to the other ants already connected to the tree under construction.

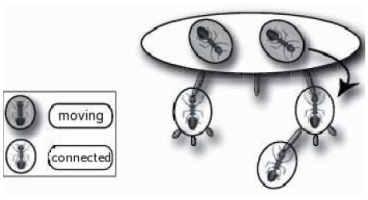


Figure 1: Tree structure generation by self-assembling artificial ants (adapted from [2]).

Each node in the tree structure represents a single ant and each ant represents a single datum. The key aspect in AntTree is the decision about where each ant will be connected, either to the main support (generating a new cluster) or to another ant (refining an existing cluster).

Each ant to be connected to the tree represents a data to be classified. Starting from an artificial support called $a_0$, all the ants will be incrementally connected either to that support or to other already connected ants. This process continues until all ants are connected to the structure, i.e., all data are already clustered. Each ant $a_i$ has associated the following terms:

1. $\mathcal{I}(a_i)$, the *ingoing links* of $a_i$. A set of links toward $a_i$ (the $a_i$'s children).
2. $\mathcal{O}(a_i)$, the *outgoing link* of $a_i$. A link to its parent node (the support or another ant).
3. A datum $d_i$ represented by $a_i$.
4. Two metrics called respectively *similarity threshold* ($T_{Sim}(a_i)$) and *dissimilarity threshold* ($T_{Dissim}(a_i)$) which will be locally updated during the process of building the tree structure.

Figure 1 shows a general outline of the self-assembling of artificial ants. It can be observed that each ant $a_i$ is either of the two following situations:

1. *Moving on the tree*: a walking ant $a_i$ (gray highlighted in figure 1) can be either on the support ($a_0$) or on another ant ($a_{pos}$). In both cases, $a_i$ is not connected to the structure. Consequently, it will be free of moving to the closest neighbors connected to either $a_0$ or $a_{pos}$. In Figure 2 is showed the neighborhood corresponding to an arbitrary ant $a_{pos}$.
2. *Connected to the tree*: in this case $a_i$ has already assigned a value for $\mathcal{O}(a_i)$, therefore, it can not move anymore. Additionally, an ant is not able to have more than $L_{max}$ ingoing links ($|\mathcal{I}(a_i)| \leq L_{max}$). The objective is to bound the maximum number of incoming links, i.e., the maximum number of clusters.

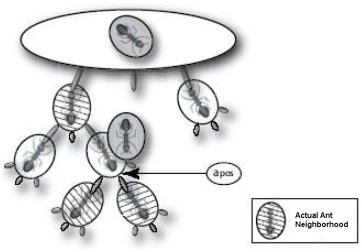


Figure 2: Neighborhood corresponding to an arbitrary ant $a_{pos}$ (adapted from [2]).

Let $\mathcal{L}$ be a list (possibly sorted) of ants to be connected
**Initialize:** Allocate all ants on the support.
$T_{Sim}(a_j) \leftarrow 1$ and $T_{Dissim}(a_j) \leftarrow 0$, for all ant $a_j$
**Repeat**
**1.** Select an ant $a_i$ from list $\mathcal{L}$
**2.** If $a_i$ is on the support ($a_0$)
then *support case* (see Figure 4)
else *ant case* (see description in [2])
**Until** all the ants are connected to the tree

Figure 3: Main loop of the AntTree algorithm.

The main loop implemented in the AntTree algorithm is shown in Figure 3. The very first step involves the allocation of all ants on the tree support and their respective thresholds of similarity and dissimilarity are accordingly initialized. In this stage, the whole collection of ants is represented by a (possibly sorted) list $\mathcal{L}$ of ants waiting to be connected in further steps. During the tree generation process each selected ant $a_i$ will be either connected to the support (or another ant) or moving on the tree looking for an adequate place to connect itself. The simulation process continue until all ants have found the more adequate assembling place; either on the support (the "Support case", see Figure 4) or on another ant (the "Ant case"). This last case is not described in the present work due to space limitations and the fact that our proposal does not affect this component of the AntTree algorithm. [1]

When $a_i$ is on the support (Figure 4) and it is the first considered ant, it is a simple situation because the ant is directly connected to the support. Otherwise, $a_i$ is compared against $a^+$, the ant most similar to $a_i$ among all the ants directly connected to the support. If these ants are similar enough, then $a_i$ will move to the subtree corresponding to $a^+$. In case that $a_i$ and $a^+$ are dissimilar enough (according to a dissimilarity threshold), $a_i$ is connected directly to the support. This last action generates a new subtree (i.e., a new cluster) due to the incoming ant is different enough from the other ants connected directly to the support. Finally, if $a_i$ is neither similar or dissimilar enough, the respective thresholds (similarity and dissimilarity) are updated in the following way: $T_{Sim}(a_i) \longleftarrow T_{Sim}(a_i)$ * 0. 9 and $T_{Dissim}(a_i) \longleftarrow T_{Dissim}(a_i)$ + 0. 01. The previous updating rules let ant $a_i$ be more "tolerant" when considered in a further iteration, i.e., the algorithm increases the probability of connecting this ant in a future time.

It is important to highlight the importance of the arrangement of the ants in the list $\mathcal{L}$ (the initial step). Since the algorithm iteratively proceeds taking the ants from $\mathcal{L}$, the features of the first ants on this list will significantly influence the final result. This will be a fundamental aspect in our proposal of the new AnTSA algorithm described in subsection 3.2.

[1] A detailed description of the "Ant case" is available in [2].

If no ant is connected to the support then connect $a_i$ to $a_0$
else
Let $a^+$ be the ant connected to $a_0$ most similar to $a_i$
**(a)** If $Sim(a_i, a^+) \geq T_{Sim}(a_i)$ then move $a_i$ toward $a^+$
**(b)** else
i. If $Sim(a_i, a^+) < T_{Dissim}(a_i)$ then connect $a_i$ to $a_0$ (in case there is no more links available in $a_0$, then move $a_i$ toward $a^+$ and decrease $T_{Sim}(a_i)$)
ii. else decrease $T_{Sim}(a_i)$ and increase $T_{Dissim}(a_i)$

Figure 4: Support case.

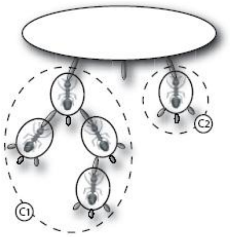


Figure 5: A tree interpreted as a non hierarchical data partition (adapted from [2]).

The resulting tree (see Figure 5) can be interpreted as a data partition (considering each ant connected to $a_0$ as a different group) as well as a dendrogram where the ants in the inner nodes could move to the leaves following the most similar nodes to them.

### 3.2 The AntSA algorithm

Some steps and processes of the AntTree method have a significative influence during the generation of the main groups. For instance, the *initial ordering step* that determines the order in which ants will be considered to be connected in the support structure (each one representing a different group) is one of those aspects. Another important process is the comparison of an arbitrary ant with the ants connected to the support (Figure 4, step **(a)**) because it determines the primary cluster assignments of ants, depending on the selected path. Our proposal basically attempts to improve the performance of AntTree by:

1. considering in the *initial step* of AntTree, additional information about the Silhouette Coefficient of previous clusterings;

2. using a more informative criterium (based on the concept of *attraction*) when the ants have to decide which path to follow in the *support case*.

**Using silhouette coefficient information in the *initial step*.** The initial ordering step defines the order in which ants will be connected to the support (each one representing a different group). Therefore, any little modification in this ordering will significatively impact the clustering results. Our proposal consists in taking as input the clustering obtained with some arbitrary clustering algorithm and using the Silhouette Coefficient (SC) information of this grouping to determine the initial order of ants. The general idea is shown in Figure 6.

**1.** Use a clustering algorithm to obtain an initial grouping.
**2.** Build $k$ data rows (one for each group obtained in the previous step) and sort them in decreasing order according to the Silhouette Coefficient.
**3.** Connect to the support the first ant of each row.
**4.** Merge the rows by iteratively taking the first ant of each non-empty row, until all rows are empty.

Figure 6: A new SC-based ordering for the AntTree's initial step.

The SC-based ordering of ants carried out in this stage determines which will be the first ants connected to the support structure. The ants with the highest $SC$ value within each group will be considered more desirable because they are the most representative ants of their groups.

***Support Case*: Attraction-based comparison.** A key aspect for an arbitrary ant $a_i$ on the support is the decision about which connected ant $a_+$ should move toward. In fact, this decision will determine the general group in which $a_i$ will be incorporated. AntTree takes into account for this decision, the similarity between $a_i$ and its most similar ant connected to the support ($a^+$). This is a "local" approach that only considers the ant directly connected to the support structure ($a^+$) but it does not take into account the ants previously connected to $a^+$, that will be denoted as $\mathcal{A}_{a^+}$. A more global approach that also considers some information on $\mathcal{A}_{a^+}$ could be useful to improve the clustering results. If $\mathcal{G}_{a^+} = \{a^+\} \cup \mathcal{A}_{a^+}$ is the group formed by $a^+$ and its descendants, this relationship between the group $\mathcal{G}_{a^+}$ and the ant $a_i$ will be referred as the *attraction of* $\mathcal{G}_{a^+}$ *on* $a_i$ and will be denoted as $att(a_i, \mathcal{G}_{a^+})$.

The idea of having different groups exerting some kind of "attraction" on the objects to be clustered was already posed in [14], where it was used as a efficient tool to obtain "dense" groups. In the present work, we will give a more general sense to the concept of attraction by considering that $att(a_i, \mathcal{G}_{a^+})$ represents *any plausible estimation of the quality of the group that would result if* $a_i$ *were incorporated to* $\mathcal{G}_{a^+}$ ($\mathcal{G}_{a^+} \cup \{a_i\}$). Thus, the only modification that AntSA will introduce to the *support case* of AntTree will be the replacement of all occurrences of $Sim(a_i, a^+)$ by $att(a_i, \mathcal{G}_{a^+})$, where $a^+$ now will represent the ant with the highest $att(a_i, \mathcal{G}_{a^+})$ value.

To compute $att(a_i, \mathcal{G}_{a^+})$ we can use some ICVM that allows to estimate the quality of individual clusters, and to apply this ICVM to $\mathcal{G}_{a^+} \cup \{a_i\}$. For instance, any *cohesion*-based ICVM could be used in this case, but other more elaborated approaches (like the density-based ones) would also be valid alternatives. As an example, an effective attraction measure is the average similarity between $a_i$ and all the ants in $\mathcal{G}_{a^+}$ as shown in Equation 1.

$$att(a_i, \mathcal{G}_{a^+}) = \frac{\sum_{a \in \mathcal{G}_{a^+}} Sim(a_i, a)}{|\mathcal{G}_{a^+}|} \qquad (1)$$

## 4 EXPERIMENTAL SETTING AND ANALYSIS OF RESULTS

For the experimental work, four collections with different levels of complexity with respect to the length of documents and vocabulary overlapping were selected: CICling-2002, EasyAbstracts, Micro4News and SEPLN-CICLing. CICling-2002 is a well known short-text collection that in different works [9, 1, 10, 8, 5, 3, 7] has been recognized as a very difficult collection since its documents are narrow domain scientific abstracts (short-length documents with a high vocabulary overlapping). Micro4News is a low complexity collection of medium-length documents about well-differentiated topics (wide domain). The EasyAbstracts corpus is composed of short-length documents (scientific abstracts) on well differentiated topics (medium complexity corpus). Finally, SEPLN-CICLing is a corpus that it is supposed to be harder to cluster than the previous corpora since its documents are narrow domain abstracts. SEPLN-CICLing and CICling-2002 have similar characteristics. However, all the SEPLN-CICLing's abstracts guarantee a minimum quality level with respect to their lengths, an aspect that is not assured by all the CICling-2002's documents.[2]

The documents were represented with the standard (normalized) $tf$-$idf$ codification after a *stop-word* removing process. The popular *cosine measure* was used to estimate the similarity between two documents. The initial data partitions required by AntSA were obtained with CLUDIPSO (using $GS$ as objective function). The parameter settings for CLUDIPSO and the remainder algorithms used in the comparison with AntSA corresponds to the parameters empirically derived in [7]. The attraction measure ($att(\cdot)$) used in our study corresponds to the formula presented in equation 1. We will refer as *AntSA-CLU* to this instance of AntSA that takes as input the CLUDIPSO's results.

[2]Space limitations prevent us from giving a more detailed description of these corpora but it is possible to obtain in [4, 9, 1, 10, 8, 5, 3, 7] more information about the features of these corpora and some links to access them for research proposes.

### 4.1 Experimental results

The results of AntSA-CLU were compared with the results obtained with other five clustering algorithms: $K$-means, CLUDIPSO [3, 7], Ant-Tree [2], MajorClust [14] and DBSCAN. $K$-means is one of the most popular clustering algorithms whereas MajorClust and DBSCAN are representative of the density-based approach to the clustering problem and have shown interesting results in similar problems. AntTree and CLUDIPSO can be considered as the "basis" of AntSA-CLU and, therefore, it would be interesting to analyze if AntSA-CLU achieves some improvements with respect to these algorithms. The results of the different algorithms were evaluated by 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. The values highlighted in bold in the different rows indicate the best obtained results.

Table 1 shows the $F_{min}$, $F_{max}$ and $F_{avg}$ values that $K$-means, MajorClust, DBSCAN, Ant-Tree, CLUDIPSO and AntSA-CLU obtained with the four collections. These results confirm the good performance that CLUDIPSO has already shown in previous works with collections of different complexity. However, in this case, AntSA-CLU not only obtained the same highest $F_{max}$ values that CLUDIPSO achieved in Micro4News, EasyAbstracts and SEPLN-CICLing. It also obtained the highest $F_{max}$ value on the CICling-2002 collection, the most difficult collection analyzed in our experiments. Another interesting aspect of AntSA-CLU, is the fact that its good performance was not limited to the $F_{max}$ values. It also outperformed the remainder algorithms in the $F_{min}$ and $F_{avg}$ values obtained with the four collections. These results show an interesting stability of AntSA-CLU which produced acceptable (or very good) results in most of the experiments. This observation can be more easily appreciated in Figure 7 where the ordered $F$-measure values obtained in the 50 experiments with AntSA-CLU (Black line) and CLUDIPSO (gray line) are displayed.

An interesting aspect to investigate is the analysis of the impact that the quality of the initial data partitioning has on the AntSA's results. An exhaustive study of this problem is beyond the scope of the present article. However, in order to get some preliminary data about this problem, we also experimented with an AntSA version that uses as input the clusterings obtained with $K$-means, the algorithm that reported the worst results in our study. We named

| | 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 |
| MajorClust | 0.90 | 0.76 | 0.96 | 0.69 | 0.44 | 0.98 | 0.59 | 0.4 | 0.77 | 0.43 | 0.37 | 0.58 |
| DBSCAN | 0.82 | 0.71 | 0.88 | 0.66 | 0.62 | 0.72 | 0.63 | 0.4 | 0.77 | 0.47 | 0.42 | 0.56 |
| AntTree | 0.7 | 0.69 | 0.82 | 0.6 | 0.5 | 0.67 | 0.49 | 0.41 | 0.64 | 0.41 | 0.38 | 0.48 |
| CLUDIPSO | 0.93 | 0.85 | **1** | 0.92 | 0.85 | **0.98** | 0.72 | 0.58 | **0.85** | 0.6 | **0.47** | 0.73 |
| AntSA-CLU | **0.96** | **0.88** | **1** | **0.96** | **0.92** | **0.98** | **0.75** | **0.63** | **0.85** | **0.61** | **0.47** | **0.75** |

Table 1: $F$-measures values.

| | 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 |
| AntSA-KM | **0.84** | **0.67** | **1** | **0.76** | **0.46** | **0.96** | **0.63** | **0.44** | **0.83** | **0.54** | **0.41** | **0.7** |

Table 2: $F$-measures values.

*AntSA-KM* this particular version of AntSA. In Table 2 and Figure 8 the results obtained with $K$-means and AntSA-KM are presented. The first observation is that although AntSA-KM is not able to achieve as good results as AntSA-CLU and CLUDIPSO obtain, it outperformed most of results obtained with DBSCAN and Ant-Tree and had, in general, a performance comparable to the MajorClust's performance. However, the comparison between the performances of AntSA-KM and $K$-means deserves special attention. As it can be clearly appreciated in Table 2 and Figure 8, AntSA-KM achieved better $F$-measure values than $K$-means on all the considered collections. The previous results provide a strong evidence that although other algorithms obtained low quality clusters, AntSA is able to improve them and obtain acceptable results. Another interesting aspect is that the AntSA algorithm would seem to be a useful general mechanism that allows to refine and improve the results of very different clustering algorithms.

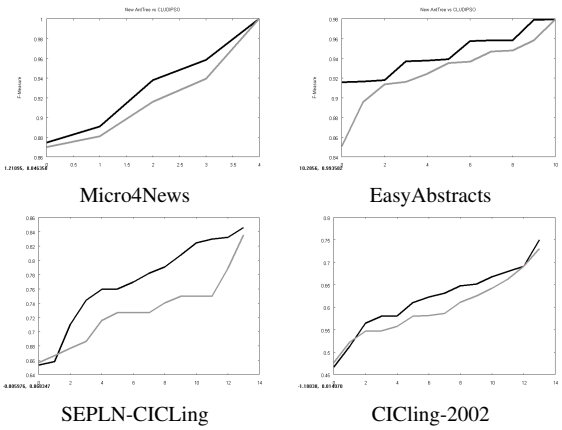


Figure 7: $F$-measure values: AntSA-CLU (Black Line) vs CLUDIPSO (Gray Line).

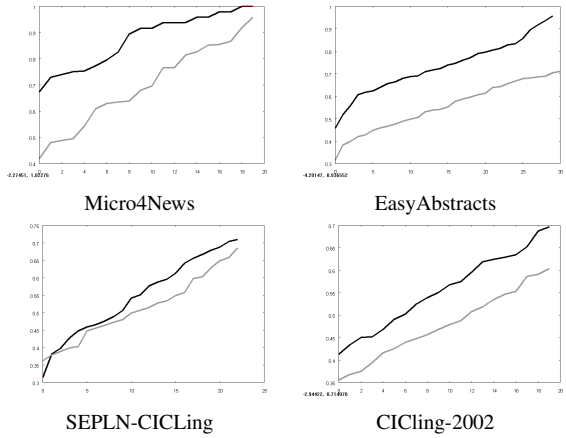


Figure 8: $F$-measure values: AntSA-KM (Black Line) vs K-Means (Gray Line).

## 5 CONCLUSIONS AND FUTURE WORK

In this work we presented AntSA, a novel AntTree-based algorithm for clustering short-text corpora. AntSA integrates information on the Silhouette coefficient and the concept of attraction in different stages of the clustering process. AntSA is a general algorithm that allows: a) to use different clustering algorithms to obtain the initial data partition, b) to define different *attraction* formulae.

When AntSA worked with the clusterings generated by the CLUDIPSO algorithm (the AntSA-CLU version), it obtained the best reported results for the four short-text collections considered in the experimental work. When the AntSA-KM version was used, it improved all the results obtained by $K$-means and had results comparable to the remainder algorithms. In all these cases, AntSA showed a significative stability in the quality of its results and presented some interesting capabilities as a general improvement method for other clustering methods.

Future work includes the use of different attraction

measures and an exhaustive experimental work that analyzes the potential improvements on other clustering algorithms. In these experiments, other more representative document collections will be considered in order to determine if the good performance of AntSA on short-text collections can also be obtained with arbitrary document collections.

## Acknowledgments

We thank the TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 research project for funding the work of the first and third authors.

## References

[1] M. Alexandrov, A. Gelbukh, and P. Rosso. An approach to clustering abstracts. In *Proc. of NLDB-05*, volume 3513 of *LNCS*, pages 8–13. Springer-Verlag, 2005.

[2] H. Azzag, N. Monmarche, M. Slimane, G. Venturini, and C. Guinot. AntTree: A new model for clustering with artificial ants. In *Proc. of the CEC2003*, pages 2642–2647, Canberra, 8-12 December 2003. IEEE Press.

[3] L. Cagnina, M. Errecalde, D. Ingaramo, and P. Rosso. A discrete particle swarm optimizer for clustering short-text corpora. In *BIOMA08*, pages 93–103, 2008.

[4] M. Errecalde and D. Ingaramo. Short-text corpora for clustering evaluation. Technical report, LIDIC, 2008.

[5] M. Errecalde, D. Ingaramo, and P. Rosso. Proximity estimation and hardness of short-text corpora. In *Proceedings of 5th Int. Workshop on Text-based Information Retrieval (TIR-2008)*, pages 15–19, 2008.

[6] D. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[7] 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.

[8] D. Ingaramo, David Pinto, P. Rosso, and M. Errecalde. Evaluation of internal validity measures in short-text corpora. In *Proc. of the CICLing 2008 Conf.*, volume 4919 of *LNCS*, pages 555–567. Springer-Verlag, 2008.

[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, 2004.

[10] D. Pinto, J. M. Benedí, and P. Rosso. Clustering narrow-domain short texts by using the Kullback-Leibler distance. In *Proc. of the CICLing 2007 Conf.*, volume 4394 of *LNCS*, pages 611–622. Springer-Verlag, 2007.

[11] D. Pinto and P. Rosso. On the relative hardness of clustering corpora. In *Proc. of TSD07*, volume 4629 of *LNAI*, pages 155–161. Springer-Verlag, 2007.

[12] Peter Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, 1987.

[13] B. Stein, S. Meyer zu Eissen, and F. Wißbrock. On cluster validity and the information need of users. In *Proc. of the IASTED03*, pages 216–221, 2003.

[14] Benno Stein and Sven Meyer zu Eißen. Document Categorization with MajorClust. In *Proc. WITS 02*, pages 91–96. Technical University of Barcelona, 2002.

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