

ITSA*: An Effective Iterative Method for Short-Text Clustering Tasks

Marcelo Errecalde¹, Diego Ingaramo¹, and Paolo Rosso²

¹ LIDIC, Universidad Nacional de San Luis, Argentina

² Natural Language Eng. Lab. ELiRF, DSIC, Universidad Polit cnica de Valencia, Spain

{merreca,daingara}@unsl.edu.ar, proso@dsic.upv.es

Abstract. The current tendency for people to use very *short documents*, e.g. blogs, text-messaging, news and others, has produced an increasing interest in automatic processing techniques which are able to deal with documents with these characteristics. In this context, “short-text clustering” is a very important research field where new clustering algorithms have been recently proposed to deal with this difficult problem. In this work, ITSA*, an iterative method based on the bio-inspired method PAntSA* is proposed for this task. ITSA* takes as input the results obtained by arbitrary clustering algorithms and refines them by iteratively using the PAntSA* algorithm. The proposal shows an interesting improvement in the results obtained with different algorithms on several short-text collections. However, ITSA* can not only be used as an effective improvement method. Using random initial clusterings, ITSA* outperforms well-known clustering algorithms in most of the experimental instances.

1 Introduction

Nowadays, the huge amount of information available on the Web offers an unlimited number of opportunities to use this information in different real life problems. Unfortunately, the automatic analysis tools that are required to make this information useful for the human comprehension, such as clustering, categorization and information extraction systems, have to face many difficulties related to the features of the documents to be processed. For example, most of Web documents like blogs, snippets, chats, FAQs, on-line evaluations of commercial products, e-mails, news, scientific abstracts and others are “*short texts*”. This is a central aspect if we consider the well-known problems that short documents usually pose to different natural language processing tasks [1,2].

During the last years, different works have recognized the importance (and complexity) of dealing with short documents, and some interesting results have been reported in *short-document clustering* tasks [1,2,3,4,5,6]. In particular, an approach that has shown an excellent performance is PAntSA* [7]. PAntSA* takes the clusterings generated by arbitrary clustering algorithms and attempts to improve them by using techniques based on the *Silhouette Coefficient* and the idea of *attraction* of a group.

Despite the significant improvements that PAntSA* has achieved on the results obtained with different algorithms and collections, some aspects of this algorithm deserve a deeper analysis. For example, it is not clear if PAntSA* would be benefitted with an iterative approach where PAntSA* is provided with a clustering generated by the own PAntSA* algorithm in the previous cycle. On the other hand, it would be interesting to analyze the performance that this iterative version of PAntSA* can achieve when the use of an additional clustering algorithm is avoided and it only takes as input randomly generated clusterings.

Our work focusses on the previous aspects by defining in the first place, an iterative version of PAntSA* that we called ITSA*. Then, an exhaustive experimental study is carried out where the performance of ITSA* as a general improvement method is compared with the performance of the original (non-iterative) PAntSA*. Finally, the performance of ITSA* as a complete clustering algorithm is evaluated, by taking as initial groupings in this case, randomly generated clusterings.

The remainder of the paper is organized as follows. Section 2 describes the main ideas of ITSA*. The experimental setup and the analysis of the results is provided in Section 3. Finally, in Section 4 some general conclusions are drawn and possible future work is discussed.

2 The ITSA* Algorithm

The *ITerative PAntSA** (*ITSA**) algorithm, is the iterative version of *PAntSA**, a bio-inspired method intended to improve the results obtained with arbitrary document clustering algorithms. PAntSA* is the partitional version of the AntTree algorithm [8] but it also incorporates information about the *Silhouette Coefficient* [9] and the concept of *attraction* of a cluster. In PAntSA*, data (in this case *documents*) are represented by *ants* which move on a tree structure according to their similarity to the other ants already connected to the tree. Each node in the tree structure represents a single ant and each ant represents a single datum.

The whole collection of ants is initially represented by a \mathcal{L} list of ants waiting to be connected. Starting from an artificial support 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.

Two main components of PAntSA* are: 1) the initial arrangement of the ants in the \mathcal{L} list, and 2) the criterium used by an arbitrary ant a_i on the support to decide which connected ant a_+ should move toward. For the first step, PAntSA* takes as input the clustering obtained with some arbitrary clustering algorithm and uses the Silhouette Coefficient (SC) information of this grouping to determine the initial order of ants in \mathcal{L} . For the second process, PAntSA* uses a more informative criterium based on the concept of *attraction*. Here, if $\mathcal{G}_{a^+} = \{a^+\} \cup \mathcal{A}_{a^+}$ is the group formed by an ant a^+ connected to the support 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 $att(a_i, \mathcal{G}_{a^+})$.

PAntSA* differs from AntTree in another central aspect: it does not build hierarchical structures which have roots (ants) directly connected to the support. In PAntSA*, each ant a_j connected to the support (a_0) and its descendants (the \mathcal{G}_{a_j} group) is considered as a simple set. In that way, when an arbitrary ant a_i has to be incorporated to the group of the ant a^+ that more attraction exerts on a_i , this step is implemented by simply adding a_i to the \mathcal{G}_{a^+} set. The resulting PAntSA* algorithm is given in Figure 1, where it is possible to observe that it takes an arbitrary clustering as input and carries out the following three steps, in order to obtain the new clustering: 1) Connection to the support, 2) Generation of the \mathcal{L} list and 3) Cluster the ants in \mathcal{L} .

In the first step, the most representative ant of each group of the clustering received as input is connected to the support a_0 . This task involves to select the ant a_i with the highest SC value of each group C_i , and to connect each one of them to the support by generating a singleton set \mathcal{G}_{a_i} . The second step consists in generating the \mathcal{L} list with the ants not connected in the previous step. This process also considers the SC-based ordering obtained in the previous step, and merges the remaining (ordered) ants of each group by iteratively taking the first ant of each non-empty queue, until all queues are empty. In the third step, the order in which these ants will be processed is determined by their positions in the \mathcal{L} list. The clustering process of each arbitrary ant a_i simply determines the connected ant a^+ which exerts more attraction on a_i ¹ and then includes a_i in the a^+ group (\mathcal{G}_{a^+}). The algorithm finally returns a clustering formed by the groups of the ants connected to the support.

Once the PAntSA* is implemented, its iterative version ITSA* can be easily obtained. ITSA* will have to provide an initial clustering to PAntSA*, take the output of PAntSA* as the new PAntSA*'s input for the next iteration, and repeat this process until no change is observed in the clustering generated by PAntSA* with respect to the previous iteration.

3 Experimental Setting and Analysis of Results

For the experimental work, seven collections with different levels of complexity with respect to the size, length of documents and vocabulary overlapping were selected: CICling-2002, EasyAbstracts, Micro4News, SEPLN-CICling, R4, R8+ and R8-. The last three corpora are subsets of the well known R8-Test corpus, a subcollection of the Reuters-21578 dataset².

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 parameter settings for CLUDIPSO and the remainder algorithms used in the comparisons correspond to the parameters empirically derived in [5].

¹ We currently use the average similarity between a_i and all the ants in \mathcal{G}_{a^+} as attraction measure but other alternatives would be also valid.

² Space limitations prevent us from giving a description of these corpora but it is possible to obtain in [10,11,12,3,2,4,5] more information about the features of these corpora and some links to access them for research proposes.

```

function PAntSA*( $\mathcal{C}$ ) returns a clustering  $\mathcal{C}^*$ 
  input:  $\mathcal{C} = \{C_1, \dots, C_k\}$ , an initial grouping
  1. Connection to the support
    1.a. Create a set  $\mathcal{Q} = \{q_1, \dots, q_k\}$  of  $k$  data queues (one queue for each
        group  $C_j \in \mathcal{C}$ ).
    1.b. Sort each queue  $q_j \in \mathcal{Q}$  in decreasing order according to the Silhouette
        Coefficient of its elements. Let  $\mathcal{Q}' = \{q'_1, \dots, q'_k\}$  be the resulting set of
        ordered queues.
    1.c. Let  $\mathcal{G}_{\mathcal{F}} = \{a_1, \dots, a_k\}$  be the set formed by the first ant  $a_i$  of each
        queue  $q'_i \in \mathcal{Q}'$ . For each ant  $a_i \in \mathcal{G}_{\mathcal{F}}$ , remove  $a_i$  from  $q'_i$  and set
         $\mathcal{G}_{a_i} = \{a_i\}$  (connect  $a_i$  to the support  $a_0$ ).
  2. Generation of the  $\mathcal{L}$  list
    2.a. Let  $\mathcal{Q}'' = \{q''_1, \dots, q''_k\}$  the set of queues resulting from the previous
        process of removing the first ant of each queue in  $\mathcal{Q}'$ .
        Generate the  $\mathcal{L}$  list by merging the queues in  $\mathcal{Q}''$ .
  3. Clustering process
    3.a. Repeat
      3.a.1 Select the first ant  $a_i$  from the list  $\mathcal{L}$ .
      3.a.2 Let  $a^+$  the ant with the highest  $att(a_i, \mathcal{G}_{a^+})$  value.
          
$$\mathcal{G}_{a^+} \leftarrow \mathcal{G}_{a^+} \cup \{a_i\}$$

    Until  $\mathcal{L}$  is empty
  return  $\mathcal{C}^* = \{\mathcal{G}_{a_1}, \dots, \mathcal{G}_{a_k}\}$ 

```

Fig. 1. The PAntSA* algorithm

3.1 Experimental Results

The results of ITSA* were compared with the results of PAntSA* and other four clustering algorithms: K -means, K -MajorClust [13], CHAMELEON [14] and CLUDIPSO [4,5]. All these algorithms have been used in similar studies and in particular, CLUDIPSO has obtained in previous works [4,5] the best results in experiments with the four small size short-text collections presented in Section 3 (CICling-2002, EasyAbstracts, Micro4News and SEPLN-CICling).

The quality of the results was 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 tables indicate the best obtained results.

Tables 1 and 2 show the F_{min} , F_{max} and F_{avg} values that K -means, K -MajorClust, CHAMELEON and CLUDIPSO obtained with the seven collections. These tables also include the results obtained with PAntSA* and ITSA* taking as input the groupings generated by these algorithms. They will be denoted with a “*” superscript for the PAntSA* algorithm and a “**” superscript for the ITSA* algorithm. Thus, for example, the results obtained with PAntSA* taking as input the groupings generated by K -Means, will be denoted as K -Means*, and those obtained with ITSA* with the same groupings as K -Means**.

Table 1. Best F -measures values per collection

	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 -Means*	0.84	0.67	1	0.76	0.46	0.96	0.63	0.44	0.83	0.54	0.41	0.7
K -Means**	0.9	0.7	1	0.94	0.71	1	0.73	0.65	0.83	0.6	0.47	0.73
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
K -MajorClust*	0.97	0.96	1	0.82	0.71	0.98	0.68	0.61	0.83	0.48	0.41	0.57
K -MajorClust**	0.98	0.97	1	0.92	0.81	1	0.72	0.71	0.88	0.61	0.46	0.73
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
CHAMELEON*	0.85	0.71	0.96	0.91	0.62	0.98	0.69	0.53	0.77	0.51	0.42	0.62
CHAMELEON**	0.93	0.74	0.96	0.92	0.67	0.98	0.69	0.56	0.79	0.59	0.48	0.71
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*	0.96	0.88	1	0.96	0.92	0.98	0.75	0.63	0.85	0.61	0.47	0.75
CLUDIPSO**	0.96	0.89	1	0.96	0.94	0.98	0.75	0.65	0.85	0.63	0.51	0.7

Table 2. Best F -measures values per collection

	R4			R8-			R8+		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K -Means	0.73	0.57	0.91	0.64	0.55	0.72	0.60	0.46	0.72
K -Means*	0.77	0.58	0.95	0.67	0.52	0.78	0.65	0.56	0.73
K -Means**	0.78	0.6	0.95	0.68	0.61	0.78	0.65	0.56	0.73
K -MajorClust	0.70	0.45	0.79	0.61	0.49	0.7	0.57	0.45	0.69
K -MajorClust*	0.7	0.46	0.84	0.61	0.5	0.71	0.63	0.55	0.72
K -MajorClust**	0.78	0.7	0.94	0.7	0.58	0.75	0.64	0.57	0.68
CHAMELEON	0.61	0.47	0.83	0.57	0.41	0.75	0.48	0.4	0.6
CHAMELEON*	0.69	0.6	0.87	0.67	0.6	0.77	0.61	0.55	0.67
CHAMELEON**	0.76	0.65	0.89	0.68	0.63	0.75	0.65	0.6	0.69
CLUDIPSO	0.64	0.48	0.75	0.62	0.49	0.72	0.57	0.45	0.65
CLUDIPSO*	0.71	0.53	0.85	0.69	0.54	0.79	0.66	0.57	0.72
CLUDIPSO**	0.74	0.53	0.89	0.7	0.63	0.8	0.68	0.63	0.74

These results are conclusive with respect to the good performance that ITSA* can obtain with short-text collections with very different characteristics. With the exception of the F_{max} value obtained with CICling-2002, it achieves the highest F_{min} , F_{avg} and F_{max} values for all the collections considered in our study, by improving the clusterings obtained with different algorithms. Thus, for instance, ITSA* obtains the highest F_{avg} value for CICling-2002 by improving the clusterings obtained by CLUDIPSO and the highest F_{avg} value for R4 by improving the clusterings obtained by K -Means and K -MajorClust. We can also appreciate in these tables that both improvement algorithms, PAntSA* and ITSA*, obtain in most of the considered cases considerable improvements on the original clusterings. Thus, for example, the F_{avg} value corresponding to the clusterings generated by K -Means with the Micro4News collection ($F_{avg} = 0.67$), is considerably improved by PAntSA* ($F_{avg} = 0.84$) and ITSA* ($F_{avg} = 0.9$).

Table 3. Results of PAntSA* vs. ITSA*

	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*	0.84	0.67	1	0.76	0.46	0.96	0.63	0.44	0.83	0.54	0.41	0.7
K -Means**	0.9	0.7	1	0.94	0.71	1	0.73	0.65	0.83	0.6	0.47	0.73
K -MajorClust*	0.97	0.96	1	0.82	0.71	0.98	0.68	0.61	0.83	0.48	0.41	0.57
K -MajorClust**	0.98	0.97	1	0.92	0.81	1	0.72	0.71	0.88	0.61	0.46	0.73
CHAMELEON*	0.85	0.71	0.96	0.91	0.62	0.98	0.69	0.53	0.77	0.51	0.42	0.62
CHAMELEON**	0.93	0.74	0.96	0.92	0.67	0.98	0.69	0.56	0.79	0.59	0.48	0.71
CLUDIPSO*	0.96	0.88	1	0.96	0.92	0.98	0.75	0.63	0.85	0.61	0.47	0.75
CLUDIPSO**	0.96	0.89	1	0.96	0.94	0.98	0.75	0.65	0.85	0.63	0.51	0.7

Table 4. Results of PAntSA* vs. ITSA*

	R4			R8-			R8+		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K -Means*	0.77	0.58	0.95	0.67	0.52	0.78	0.65	0.56	0.73
K -Means**	0.78	0.6	0.95	0.68	0.61	0.78	0.65	0.56	0.73
K -MajorClust*	0.70	0.46	0.84	0.61	0.5	0.71	0.63	0.55	0.72
K -MajorClust**	0.78	0.7	0.94	0.7	0.58	0.75	0.64	0.57	0.68
CHAMELEON*	0.69	0.6	0.87	0.67	0.6	0.77	0.61	0.55	0.67
CHAMELEON**	0.76	0.65	0.89	0.68	0.63	0.75	0.65	0.6	0.69
CLUDIPSO*	0.71	0.53	0.85	0.69	0.54	0.79	0.66	0.57	0.72
CLUDIPSO**	0.74	0.53	0.89	0.7	0.63	0.8	0.68	0.63	0.74

Tables 3 and 4 compare the results obtained with ITSA* with respect to PAntSA*. Here, the benefits of the iterative approach on the original improvement method are evident, with a better performance of ITSA* on PAntSA* in most of the considered experimental instances. As an example, when ITSA* took as input the clusterings generated by K -Means, its results were consistently better than (or as good as) those obtained by PAntSA* with the same clusterings, on the seven considered collections.

Despite the excellent results previously shown by ITSA*, it is important to observe that in a few experimental instances, ITSA* do not improve (and it can even deteriorate) the results obtained with PAntSA* or the initial clusterings generated by the other algorithms. This suggests that, despite the *average* improvements that ITSA* achieves on all the considered algorithms, and the highest F_{min} and F_{max} value obtained on most of the considered collections, a deeper analysis is required in order to also consider the *improvements* (or the *deteriorations*) that ITSA* carries out on each clustering that it receives as input.

The previous observations pose some questions about *how often* (and in *what extent*) we can expect to observe an improvement in the quality of the clusterings provided to ITSA*. Tables 5 and 6 give some insights on this subject, by

presenting in Table 5 the *improvement percentage* (*IP*) and the *improvement magnitude* (*IM*) obtained with ITSA*, whereas Table 6 gives the *deterioration percentage* (*DP*) and the *deterioration magnitude* (*DM*) that ITSA* produced on the original clusterings. The *percentage* of cases where ITSA* produces clusterings with the *same quality* as the clusterings received as input (*SQP*) can be directly estimated from the two previous percentages. Thus, for example, ITSA* produced an improvement in the 92% of the cases when received the clusterings generated by CHAMELEON on the R4 collection, giving *F*-measures values which are (on average) a 0.17 higher than the *F*-measures values obtained with CHAMELEON. In this case, *DP* = 7% and *DM* = 0.02 meaning that in 1% of the experiments with this algorithm and this collection, ITSA* gave results of the same quality (*SQP* = 1%).

Table 5. *IP* and *MP* values of ITSA* with respect to the original algorithms

	4MNG		Easy		SEPLN-CIC		CIC-2002		R4		R8-		R8+	
Algorithms	<i>IP</i>	<i>MP</i>	<i>IP</i>	<i>MP</i>	<i>IP</i>	<i>MP</i>	<i>IP</i>	<i>MP</i>	<i>IP</i>	<i>MP</i>	<i>IP</i>	<i>MP</i>	<i>IP</i>	<i>MP</i>
K-Means	100	0.3	100	0.44	100	0.27	100	0.18	96	0.06	90	0.06	88	0.07
K-MajorClust	100	0.48	100	0.49	100	0.32	100	0.22	100	0.39	100	0.29	100	0.27
CHAMELEON	100	0.18	83	0.15	50	0.16	92	0.13	92	0.17	71	0.14	100	0.18
CLUDIPSO	100	0.03	100	0.05	55	0.03	62	0.04	98	0.1	100	0.39	100	0.43

Table 6. *DP* and *DM* values of ITSA* with respect to the original algorithms

	4MNG		Easy		SEPLN-CIC		CIC-2002		R4		R8-		R8+	
Algorithms	<i>DP</i>	<i>DM</i>	<i>DP</i>	<i>DM</i>	<i>DP</i>	<i>DM</i>	<i>DP</i>	<i>DM</i>	<i>DP</i>	<i>DM</i>	<i>DP</i>	<i>DM</i>	<i>DP</i>	<i>DM</i>
K-Means	0	0	0	0	0	0	0	0	4	0.03	10	0.03	12	0.02
K-MajorClust	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CHAMELEON	0	0	16	0.05	50	0.04	7	0.03	7	0.02	28	0.001	0	0
CLUDIPSO	0	0	0	0	44	0.03	38	0.01	1	0.01	0	0	0	0

With the exception of the CHAMELEON - SEPLN-CICling and CLUDIPSO - SEPLN-CICling combinations, where ITSA* does not obtain significant improvements, the remaining experimental instances are conclusive about the advantages of using ITSA* as a general improvement method. Thus, for example, in 17 experimental instances (algorithm-collection combinations) ITSA* obtained an improvement in the 100% of the experiments. This excellent performance of ITSA* can be easily appreciated in Figure 2, where the *IP* (white bar)), *DP* (black bar) and *SQP* (gray bar) values are compared but considering in this case the improvements/deteriorations obtained in each one of the seven collections. Space limitations prevent us from showing a similar comparison of the *IP*, *DP* and *SQP* values of ITSA* with respect to PAntSA*, but in the last section some summarized data will be given about this aspect.

Finally, an important aspect to consider is to what extent the effectiveness of ITSA* depends on the quality of the clusterings received as input. In order

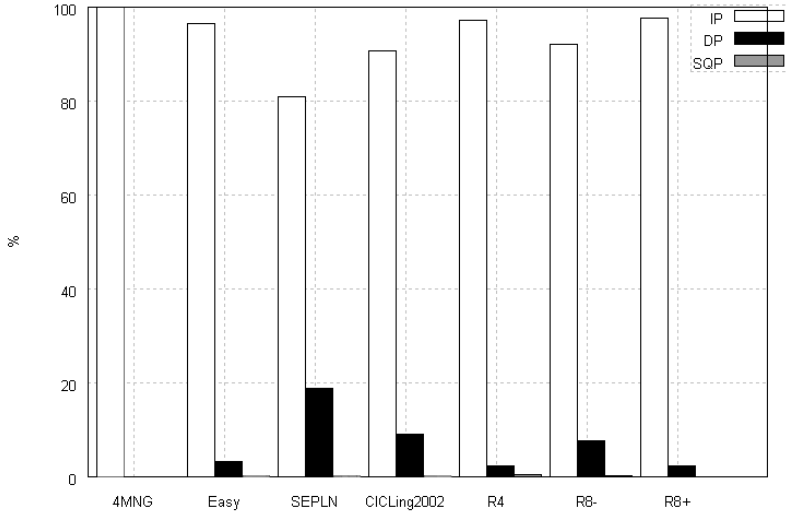


Fig. 2. *IP*, *DP* and *SQP* values per collection of ITSA* with respect to the original algorithms

Table 7. Best *F*-measures values per collection (subsets of R8-Test corpus)

	R4			R8-			R8+		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
<i>R</i> -Clustering	0.32	0.29	0.35	0.21	0.19	0.24	0.21	0.2	0.24
<i>R</i> -Clustering*	0.68	0.48	0.87	0.63	0.52	0.73	0.64	0.54	0.7
<i>R</i> -Clustering**	0.75	0.54	0.94	0.66	0.57	0.74	0.65	0.57	0.72
<i>K</i> -Means	0.73	0.57	0.91	0.64	0.55	0.72	0.60	0.46	0.72
<i>K</i> -MajorClust	0.70	0.45	0.79	0.61	0.49	0.7	0.57	0.45	0.69
CHAMELEON	0.61	0.47	0.83	0.57	0.41	0.75	0.48	0.4	0.6
CLUDIPSO	0.64	0.48	0.75	0.62	0.49	0.72	0.57	0.45	0.65

Table 8. Best *F*-measures values per collection (other corpora)

	Micro4News			EasyAbstracts			SEPLN-CILing			CILing-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}
<i>R</i> -Clustering	0.38	0.31	0.5	0.38	0.32	0.45	0.38	0.3	0.47	0.39	0.31	0.52
<i>R</i> -Clustering*	0.87	0.73	1	0.76	0.54	0.96	0.63	0.48	0.77	0.54	0.42	0.71
<i>R</i> -Clustering**	0.9	0.73	1	0.92	0.67	1	0.73	0.65	0.84	0.6	0.46	0.75
<i>K</i> -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
<i>K</i> -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

to analyze this dependency, 50 clusterings generated by a simple process that randomly determines the group of each document (denoted as *R*-Clustering) were considered. These clusterings were given as input to PAntSA* and ITSA* as initial orderings and the results were compared with the remaining algorithms considered in our experimental work. The results of these experiments are shown in Tables 8 and 7 where it is possible to appreciate that ITSA* is very robust to low quality initial clusterings. In fact, ITSA* obtains in several collections the best F_{min} , F_{max} or F_{avg} values and, in the remaining cases, it achieves results comparable to the best results obtained with other algorithms.

4 Conclusions and Future Work

In this work we presented ITSA*, an effective iterative method for the short-text clustering task. ITSA* achieved the best *F*-measure values on most of the considered collections. These results were obtained by improving the clusterings generated by different clustering algorithms.

PAntSA* does not guarantee an improvement of all the clusterings received as input. However, some data about the improvements obtained with ITSA* are conclusive with respect to its strengths as a general improvement method: on a total of 1750 experiments ($1750 = 7$ collections \times 5 algorithms \times 50 runs per algorithm) ITSA* obtained 1639 improvements, 5 results with the same quality and only 106 lower quality results with respect to the initial clusterings provided to the algorithm. On the other hand, if ITSA* is compared with the results obtained with PAntSA*, it can be seen that ITSA* obtained 1266 improvements, 219 results with the same quality and only 265 lower quality results.

An interesting result of our study is the robustness of ITSA* to low quality initial clusterings received as input. In that sense, the results obtained with random initial clusterings shown similar (or better) *F*-measure values than those obtained by well-known algorithms used in this area. This last aspect deserves a deeper analysis which will be carried out in future works.

Acknowledgments. The first and third authors have collaborated in the framework of the TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 research project. We also thank the ANPCyT and UNSL (Argentina) for funding the work of the first and second authors.

References

1. Pinto, D., Rosso, P.: On the relative hardness of clustering corpora. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 155–161. Springer, Heidelberg (2007)
2. Errecalde, M., Ingaramo, D., Rosso, P.: Proximity estimation and hardness of short-text corpora. In: Proceedings of TIR-2008, pp. 15–19. IEEE CS, Los Alamitos (2008)

3. Ingaramo, D., Pinto, D., Rosso, P., Errecalde, M.: Evaluation of internal validity measures in short-text corpora. In: Gelbukh, A. (ed.) *CICLing 2008*. LNCS, vol. 4919, pp. 555–567. Springer, Heidelberg (2008)
4. Cagnina, L., Errecalde, M., Ingaramo, D., Rosso, P.: A discrete particle swarm optimizer for clustering short-text corpora. In: *BIOMA08*, pp. 93–103 (2008)
5. Ingaramo, D., Errecalde, M., Cagnina, L., Rosso, P.: Particle Swarm Optimization for clustering short-text corpora. In: *Computational Intelligence and Bioengineering*, pp. 3–19. IOS press, Amsterdam (2009)
6. Ingaramo, D., Errecalde, M., Rosso, P.: A new anttree-based algorithm for clustering short-text corpora. *Journal of CS&T* (in press, 2010)
7. Ingaramo, D., Errecalde, M., Pinto, D.: A general bio-inspired method to improve the short-text clustering task. In: *Proc. of CICLing 2010*. LNCS. Springer, Heidelberg (in press 2010)
8. Azzag, H., Monmarche, N., Slimane, M., Venturini, G., Guinot, C.: AntTree: A new model for clustering with artificial ants. In: *Proc. of the CEC 2003*, Canberra, pp. 2642–2647. IEEE Press, Los Alamitos (2003)
9. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20, 53–65 (1987)
10. Makagonov, P., Alexandrov, M., Gelbukh, A.: Clustering abstracts instead of full texts. In: Sojka, P., Kopeček, I., Pala, K. (eds.) *TSD 2004*. LNCS (LNAI), vol. 3206, pp. 129–135. Springer, Heidelberg (2004)
11. Alexandrov, M., Gelbukh, A., Rosso, P.: An approach to clustering abstracts. In: Montoyo, A., Muñoz, R., Métails, E. (eds.) *NLDB 2005*. LNCS, vol. 3513, pp. 8–13. Springer, Heidelberg (2005)
12. Pinto, D., Benedí, J.M., Rosso, P.: Clustering narrow-domain short texts by using the Kullback-Leibler distance. In: Gelbukh, A. (ed.) *CICLing 2007*. LNCS, vol. 4394, pp. 611–622. Springer, Heidelberg (2007)
13. Stein, B., Meyer zu Eißén, S.: Document Categorization with MAJORCLUST. In: *Proc. WITS 02*, pp. 91–96. Technical University of Barcelona (2002)
14. Karypis, G., Han, E.H., Vipin, K.: Chameleon: Hierarchical clustering using dynamic modeling. *Computer* 32, 68–75 (1999)