A general bio-inspired method to improve the short-text clustering task

Diego Ingaramo[1] Marcelo Luis Errecalde[1] Paolo Rosso[2]

[1]Universidad Nacional de San Luis, Argentina
[2]Universidad Politécnica de Valencia, España

11th International Conference on Intelligent Text Processing and Computational Linguistics, 2010

Outline

1. **Introduction**
   - Context of our work
   - Motivations of our work
2. **A general improvement technique: The PAntSA* algorithm**
   - Main concepts
   - Using Silhouette Coefficient information
   - Attraction-based comparison
   - A partitional simplified version of AntSA
3. **Experiments**
   - Data sets
   - Algorithms
   - Test and Results

What is Document Clustering?

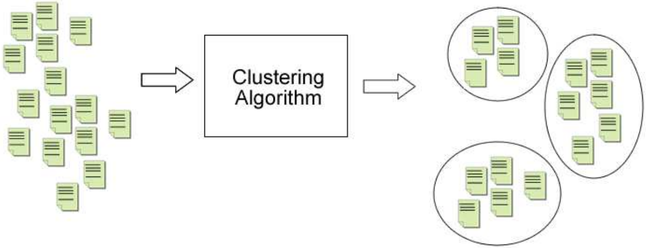
- Finding groups of documents such that the documents in a group will be similar (or related) to one another and different from (or unrelated to) the documents in other groups

What is the problem we are working on?

1. Main goal: to develop effective algorithms for the problem of clustering short-text corpora.

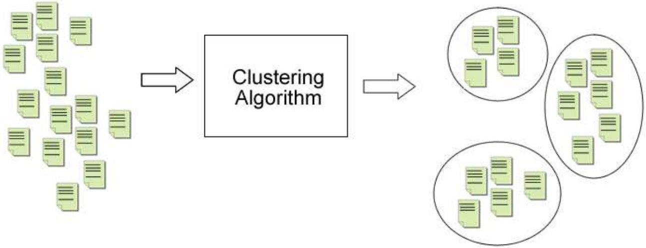
What is the problem we are working on?

1. Main goal: to develop effective algorithms for the problem of clustering short-text corpora.
2. These algorithms assign documents to unknown categories in an unsupervised way.



What is the problem we are working on?

1. Main goal: to develop effective algorithms for the problem of clustering short-text corpora.
2. These algorithms assign documents to unknown categories in an unsupervised way.



3. Our interest is on clustering of:
   - short-texts (in general)
   - narrow domain short-texts (in particular)

Why is it important?

- Applicability in different areas of text processing:
  - text mining
  - summarization
  - information retrieval
  - ...
- Tendencies of people to use ‘small-languages’:
  - blogs
  - text-messages
  - snippets
  - ...

Why is this problem difficult?

1. General problems of text clustering:
    - Synonymy.
    - Polysemy.
2. Additional difficulties due to:
    - Low frequencies of the document terms.
    - High overlapping degree of their vocabularies.

These aspects can negatively affect the estimation of how similar the documents are and (in consequence) the whole clustering process

Antecedents of our proposal: AntSA-CLU

- Our previous approach to deal with these difficult problems was a bio-inspired clustering algorithm named as AntSA-CLU(AntTree-Silhouette-Attraction).

Antecedents of our proposal: AntSA-CLU

- Our previous approach to deal with these difficult problems was a bio-inspired clustering algorithm named as AntSA-CLU(AntTree-Silhouette-Attraction).
- AntSA-CLU is a hierarchical AntTree-based algorithm which incorporates two main concepts:
  - The Silhouette Coefficient.
  - The idea of attraction of a cluster.

Antecedents of our proposal: AntSA-CLU

- Our previous approach to deal with these difficult problems was a bio-inspired clustering algorithm named as AntSA-CLU(AntTree-Silhouette-Attraction).
- AntSA-CLU is a hierarchical AntTree-based algorithm which incorporates two main concepts:
  - The Silhouette Coefficient.
  - The idea of attraction of a cluster.
- It takes as input the results of the CLUDIPSO algorithm and attempts to improve them with the new AntTree based method.

Some limitations of our work with AntSA-CLU

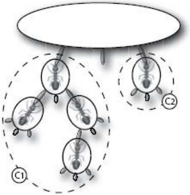
1. As initial data partitions, only the results generated by the CLUDIPSO algorithm were considered.
2. Experiments were limited to small size collections with different complexity levels.

What questions are we trying to answer in our work?

1. Can these ideas used in AntSA-CLU be successfully applied in other arbitrary algorithms?
2. Is the AntSA-CLU effectiveness limited to small size collections or it can be an useful algorithm for arbitrary size short-text collections?

Some general concepts on AntSA-CLU

- Based on the AntTree algorithm.
- Starting from an artificial support called a_0, all the ants are incrementally connected either to that support or to other ant.
- Ants move in the structure according to its similarity to the other ants already connected to the tree under construction.
- Each ant represents a single datum from the data set



The process continues until all ants have found the more adequate place, either on the support or on another ant.

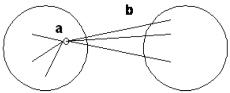
Some general concepts on AntSA-CLU

AntSA differs from AntTree in two main steps:

- The initial ordering step of ants (it incorporates information related to the Silhouette Coefficient).
- Using a more informative criterium when the ants have to decide which path to follow (concept of attraction)

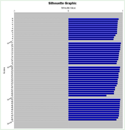
The Silhouette Coefficient (SC)

Combine ideas of both cohesion and separation.

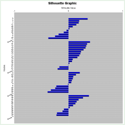


We can calculate the SC value for a particular datum (document), a cluster or the whole clustering.

Good Clustering

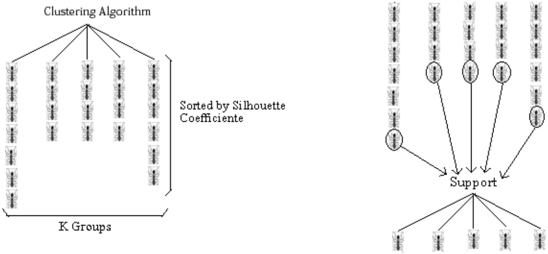


Bad Clustering



The SC in the initial ordering step

- The initial ordering step defines which ants will be connected to the support(representing a group).
- Using the Silhouette Coefficient (SC) information from a clustering obtained by an arbitrary clustering algorithm to determine the initial order of ants.
- The SC-based ordering of ants carried out in this stage determines which will be the first ants connected to the support structure.



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.

Attraction-based comparison

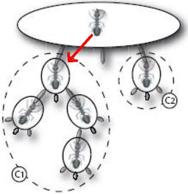
- A key aspect for an arbitrary ant a_i on the support is the decision about which connected ant a_+ should move toward.
- This decision will determine the group in which a_i will be incorporated.

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.
- This decision will determine the group in which a_i will be incorporated.
- AntTree makes this decision by considering the ant connected to the support (a_+) which is more similar to the ant being considered (a_i).

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.
- This decision will determine the group in which a_i will be incorporated.
- AntTree makes this decision by considering the ant connected to the support (a_+) which is more similar to the ant being considered (a_i).



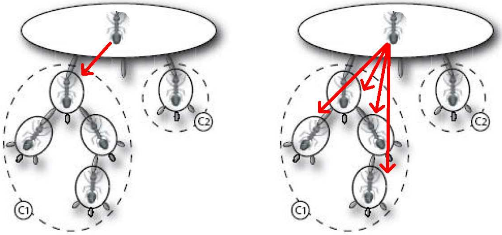
Attraction-based comparison

Unlike AntTree, we now have different groups exerting some kind of attraction on the objects to be clustered.

Attraction-based comparison

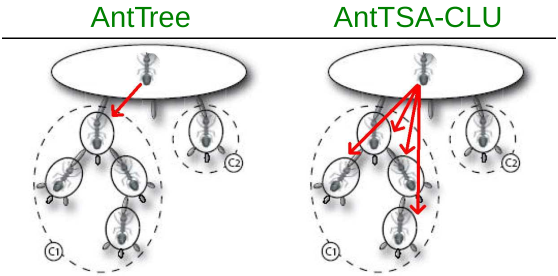
Unlike AntTree, we now have different groups exerting some kind of attraction on the objects to be clustered.

AntTree	AntTSA-CLU



Attraction-based comparison

Unlike AntTree, we now have different groups exerting some kind of attraction on the objects to be clustered.



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

PAntSA*

- When a hierarchical organization of the results is not required, some parameters and initialization steps required by AntSA are not necessary.

PAntSA*

- When a hierarchical organization of the results is not required, some parameters and initialization steps required by AntSA are not necessary.
- Removing these aspects, results in a partitioned version of AntSA, named PAntSA*.

PAntSA*

- When a hierarchical organization of the results is not required, some parameters and initialization steps required by AntSA are not necessary.
- Removing these aspects, results in a partitioned version of AntSA, named PAntSA*.
- PAntSA* is mucho more simpler and efficient than the original AntSA algorithm.

PAntSA*

- When a hierarchical organization of the results is not required, some parameters and initialization steps required by AntSA are not necessary.
- Removing these aspects, results in a partitioned version of AntSA, named PAntSA*.
- PAntSA* is mucho more simpler and efficient than the original AntSA algorithm.
- The resulting PAntSA* carries out the following three steps, in order to obtain the new clustering:
  1. Connection to the support.
  2. Generation of the L list.
  3. Cluster the ants in L.

Main algorithm

function PAntSA*($\mathcal{C}$) **returns** a clustering $\mathcal{C}^\star$
input: $\mathcal{C} = \{C_1, \ldots, C_k\}$, an initial grouping

1. Connection to the support

2. Generation of the $\mathcal{L}$ list

3. Clustering process

return $\mathcal{C}^\star = \{\mathcal{G}_{a_1}, \ldots, \mathcal{G}_{a_k}\}$

Main algorithm

function PAntSA*($\mathcal{C}$) **returns** a clustering $\mathcal{C}^\star$
input: $\mathcal{C} = \{C_1, \ldots, C_k\}$, an initial grouping

1. Connection to the support

1.a. Create a set $\mathcal{Q} = \{q_1, \ldots, q_k\}$ 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 $\mathcal{Q}' = \{q'_1, \ldots, q'_k\}$ be the resulting set of ordered queues.

1.c. Let $\mathcal{G}_\mathcal{F} = \{a_1, \ldots, a_k\}$ 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 $\mathcal{Q}'' = \{q''_1, \ldots, q''_k\}$ the set of queues resulting from the previous process of removing the first ant of each queue in $\mathcal{Q}'$.
Generate a $\mathcal{L}$ 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 $\mathcal{C}^\star = \{\mathcal{G}_{a_1}, \ldots, \mathcal{G}_{a_k}\}$

Small short-texts collections

We select four short-text collection to test our approach:

- CICling-2002: considered in many research works, is a high complexity corpus, with short-length documents and high vocabulary overlapping.
- SEPLN-CICLing: collection with short-length documents, easier than CICling-2002 with respect to the length of documents.
- EasyAbstracts: collection easier than SEPLN-CICLing with respect to the overlapping degree of the documents vocabulary.
- Micro4News: collection with medium-length documents, the most easy collection with respect to the length of documents and vocabulary overlapping.

Reuters based collections

We present three new based Reuters collection to test our approach:

- R8+: is a high complexity corpus, with eight unbalanced groups and high vocabulary overlapping.
- R8-: is a medium complexity corpus, with eight unbalanced groups and low vocabulary overlapping.
- R4: the most easy Reuters collection generated with low vocabulary overlapping and four balanced groups.

Test algorithms

The results of PAntSA* were compared with the results of other four clustering algorithms:

Test algorithms

The results of PAntSA* were compared with the results of other four clustering algorithms:

- K-Means
- K-MajorClust
- CHAMELEON
- CLUDIPSO

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.

	Micro4News			EasyAbstracts		
Algorithms	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
K-Means*	0.84	0.67	**1**	0.76	0.46	0.96
K-MajorClust	0.95	0.94	0.96	0.71	0.48	**0.98**
K-MajorClust*	**0.97**	**0.96**	**1**	0.82	0.71	**0.98**
CHAMELEON	0.76	0.46	0.96	0.74	0.39	0.96
CHAMELEON*	0.85	0.71	0.96	0.91	0.62	**0.98**
CLUDIPSO	0.93	0.85	**1**	0.92	0.85	**0.98**
CLUDIPSO*	0.96	0.88	**1**	**0.96**	**0.92**	**0.98**

	SEPLN-CICLing			CICling-2002		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means	0.49	0.36	0.69	0.45	0.35	0.6
K-Means*	0.63	0.44	0.83	0.54	0.41	0.7
K-MajorClust	0.63	0.52	0.75	0.39	0.36	0.48
K-MajorClust*	0.68	0.61	0.83	0.48	0.41	0.57
CHAMELEON	0.64	0.4	0.76	0.46	0.38	0.52
CHAMELEON*	0.69	0.53	0.77	0.51	0.42	0.62
CLUDIPSO	0.72	0.58	**0.85**	0.6	**0.47**	0.73
CLUDIPSO*	**0.75**	**0.63**	**0.85**	**0.61**	**0.47**	**0.75**

Table: 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-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
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
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

Table: Best F-measures values per collection.

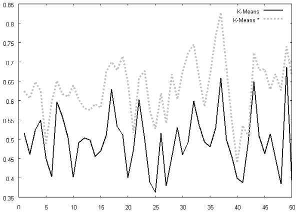
	Micro4News			EasyAbstracts		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means	0.67	0.41	0.96	0.54	0.31	0.71
K-Means*	**0.84**	**0.67**	**1**	**0.76**	**0.46**	**0.96**
K-MajorClust	0.95	0.94	0.96	0.71	0.48	0.98
K-MajorClust*	**0.97**	**0.96**	**1**	**0.82**	**0.71**	**0.98**
CHAMELEON	0.76	0.46	**0.96**	0.74	0.39	0.96
CHAMELEON*	**0.85**	**0.71**	**0.96**	**0.91**	**0.62**	**0.98**
CLUDIPSO	0.93	0.85	**1**	0.92	0.85	**0.98**
CLUDIPSO*	**0.96**	**0.88**	**1**	**0.96**	**0.92**	**0.98**

	SEPLN-CICLing			CICling-2002		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means	0.49	0.36	0.69	0.45	0.35	0.6
K-Means*	**0.63**	**0.44**	**0.83**	**0.54**	**0.41**	**0.7**
K-MajorClust	0.63	0.52	0.75	0.39	0.36	0.48
K-MajorClust*	**0.68**	**0.61**	**0.83**	**0.48**	**0.41**	**0.57**
CHAMELEON	0.64	0.4	0.76	0.46	0.38	0.52
CHAMELEON*	**0.69**	**0.53**	**0.77**	**0.51**	**0.42**	**0.62**
CLUDIPSO	0.72	0.58	**0.85**	0.6	**0.47**	0.73
CLUDIPSO*	**0.75**	**0.63**	**0.85**	**0.61**	**0.47**	**0.75**

Table: Results of PAntSA* vs. groupings generated by different

	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-MajorClust	**0.70**	0.45	0.79	**0.61**	0.49	0.7	0.57	0.45	0.69
K-MajorClust*	**0.70**	**0.46**	**0.84**	**0.61**	**0.5**	**0.71**	**0.63**	**0.55**	**0.72**
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**
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**

Table: Results of PAntSA* vs. groupings generated by different algorithms.



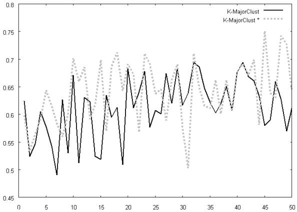


Figure: Results of PAntSA* with a significant (left) and a minor (right) improvement level.

	4MNG		Easy		SEPLN-CIC		CIC-2002	
Algorithms	*IP*	*MP*	*IP*	*MP*	*IP*	*MP*	*IP*	*MP*
K-Means	94	0.18	94	0.24	100	0.14	96	0.09
K-MajorClust	50	0.03	94	0.13	94	0.04	100	0.09
CHAMELEON	87	0.11	100	0.17	100	0.07	75	0.08
CLUDIPSO	74	0.05	86	0.05	84	0.03	92	0.03

	R4		R8-		R8+	
Algorithms	*IP*	*MP*	*IP*	*MP*	*IP*	*MP*
K-Means	56	0.1	70	0.07	97	0.07
K-MajorClust	96	0.05	61	0.06	97	0.07
CHAMELEON	91	0.09	85	0.1	100	0.13
CLUDIPSO	76	0.12	84	0.09	89	0.06

Table: *IP* and *MP* values

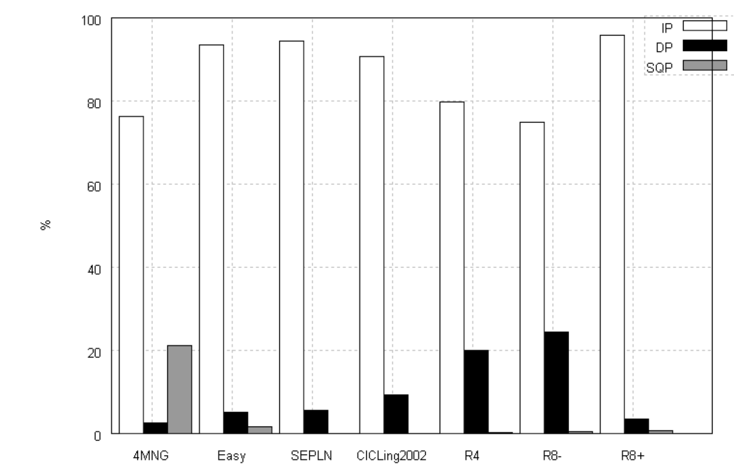


Figure: *IP*, *DP* and *SQP* values per collection.

Conclusions

- We presented PAntSA*, a general bio-inspired method to improve the short-text clustering task.
- PAntSA* achieved the best F_{min}, F_{max} and F_{avg} values on all the considered collections.
- A decrease in the F-measure values of the results produced by PAntSA* is not a very frequent result.

Future work

- Provide to PAntSA* with a clustering generated by the own PAntSA* algorithm.
- Test this improvement with random initial clusterings.

Questions?

Questions?

Thank You very much for your attention...