A New Multi-layered Approach for Automatic Text Summaries Mono-Document Based on Social Spiders

Mohamed Amine Boudia, Reda Mohamed Hamou, Abdelmalek Amine, Mohamed Elhadi Rahmani, Amine Rahmani

A new multi-layered approach for automatic text summaries Mono-Document based on social spiders

Mohamed Amine BOUDIA [1], Reda Mohamed HAMOU [2], Abdelmalek AMINE [3], Mohamed Elhadi RAHMANI[4], Amine RAHMANI[5]

Dr. Moulay Tahar University SAÏDA
Department of Computer Saida, Algeria
Laboratory Knowledge Management and Complex Data (GeCoDe Lab)
{mamiamounti[1], hamoureda[2], abd_amine1[3]}@yahoo.fr
r_m_elhadi@yahoo.fr[4], aminerahmani2091@gmail.com[5]

Abstract. In this paper, we propose a new multi layer approach for automatic text summarization by extraction where the first layer constitute to use two techniques of extraction : scoring of phrases, and similarity that aims to eliminate redundant phrases without losing the theme of the text. While the second layer aims to optimize the results of the previous layer by the metaheuristic based on social spiders. the objective function of the optimization is to maximize the sum of similarity between phrases of the candidate summary in order to keep the theme of the text, minimize the sum of scores in order to increase the summarization rate, this optimization also will give a candidate's summary where the order of the phrases changes compared to the original text.The third and final layer aims to choose the best summary from the candidate summaries generated by layer optimization, we opted for the technique of voting with a simple majority.

Keywords: Automatic Summary Extraction, Data Mining, Social Spider, optimization, Scoring, similarity

1 Introduction and problematic

Every day, the mass of electronic textual information is increasing, making it more and more difficult access to relevant information without using specific tools. In other words access to the content of the texts by rapid and effective ways is becoming a necessity.

A summary of a text is an effective way to represent its contents, and allow quick access to their semantic content. The purpose of a summarization is to produce an abridged text covering most of the content from the source text.

« We can not imagine our daily life, one day without summaries », underline Inderjeet Mani. Newspaper headlines, the first paragraph of a newspaper article, newsletters, weather, tables of results of sports competitions and library catalogs are all sum-

marized. Even in the research, the authors of scientific articles must accompany their scientific articles by a summary written by themselves.

Automatic summary can be used to reduce the search time to find the relevant documents or to reduce the treatment of long texts by identifying the key information.

Our work uses automatic summarization by extraction, because it is a simple method to implement and gives good results; only in the previous works, produce the automatic summary by extraction consists to use only one technique at a time (Score, Similarity sentence or prototype) and respects the order of the sentences in the original document, our work answers the following questions:

- What is the contribution of the use of two methods of summarization at the same time on the quality of summary?
- Can the bio-inspired method based on the social spiders brings more for the automatic summary and increase the quality of the summary?

2 Our proposed approach

To create a summary by extraction, it is necessary to identify textual units (phrases, clauses, sentences, paragraphs) considered salient (relevant), then the select the textual units that hold the main ideas of the text with a certain order, in order to build a summary.

The approach presented in this article obeys the following steps:

2.1 Pretreatment

Simple cleaning: a stop words will not be removed, because the method of automatic summarization by extraction aims to extract the most informative sentences without modifying them: if we remove the empty words without information on their morpho-syntactic impact in sentences, we risk having an inconsistent summary of a morphological point of view.

Then cleaning is to remove emoticons, to replace spaces with "_" and remove special characters (#, \, [,] .....).

– Choice of term: for automatic summarization by extraction we will need two representations:

- Bag of words representation.
- Bag of sentence representation.

Both representations are introduced in the vector model.

The first representation is to transform the text into a vector v_i (w_1, w_2, ..., $w_{|T|}$) where T is the number of all the words that appear at least once in the text. The weight w_k indicates the occurrence of t_k word in the document.

The second representation is to transform the text into a V'I vector (q_1, q_2, ..., $q_{|R|}$) where R is the number of all the phrases that appear at least once in the text. The q_k weight indicates the occurrence of t_k sentence in the document.

And finally a word phrases- occurrence matrix will be generated after the two previous representation, the size of this matrix is equal to (the number of words in the text) X (the number of words in the text); p_{ik} weight is the number occurrence of the word *i* in the sentence *j*;

2.2 Layer 1 : pre-summary

Weighting and pre-summary.

Weighting.
Once the "Word-Phrase" matrix is ready, we calculate a weighting of ""Word-Phrase" matrix using a known encodings (tf-idf or tfc) with a small modification to the adapted the concept of a mono-document summarization.

The weight of a term in a sentence t_k p_i is calculated as:

- *TF-IDF*.

$$tf - idf(t_k, p_i) = tf(t_k, p_i) * \log(\frac{A}{B}) \quad (1)$$

$tf(t_k, p_i)$: the number of occurrences of the term tk in the phrase pi;
A : the total number of sentences in the text;
B : the number of sentences in which the tk term appears at least once.

- *TFC*.

$$\text{tfc}(t_k, p_i) = \frac{\text{tf-idf}(t_k, p_i)}{\sqrt{\sum_{i=1}^{|p|} \text{tf-idf}(t_k, p_i)^2}} \quad (2)$$

After calculating the weighting of each word, a weight is assigned to each sentence. The generated summary is then generated by displaying the highest score of the source document sentences.

This score of a sentence is equal to the sum of the words in this sentence:

$$\text{SCORE}(p_i) = \sum_{k=0}^{nbr\_word} Mik \quad (3)$$

Primitive summary.

"Suggested process claims on the principle that high-frequency words in a document are important words" [Luhn 1958]

The final step is to select the N first sentences that have the highest weight and which are considered the most relevant. The process of extracting the first N sentences intended to build the summary is defined either by a threshold, in this case, the score of the sentence must be greater than or equal to the threshold in order that this sentence will be extracted the second method is to fix a number N of phrase to be extracted, all phases will be ranked in descending order according of their score, and we take only the first N phrases.

Elimination of rehearsals and theme detection: using SIMILARITY method summarization by extraction.

The result of the previous step is a set of phrases which is a high score. Just we have a possibility that two or more sentences have a high score but they are similar, so we proceed to the elimination of phrases that resembling. The similarity between the sentences that have been selected at the end of the previous step with known metrics (Euclidean .....).

Two parameters are used to adjust the elimination of repetitions: similarity threshold and reduction rate, the first parameter defined the point that we can consider two sentences as similar, and the second parameter indicates the number of resemblance to eliminate, to decrease the entropy information. When the similarity between two sentences is greater: they the phrase that has the highest score stay and we remove the other sentence.

The similarity is also used to detect the sentence that has more relation with the them of the text. According to the domain experts, it is the sentence which is most similar to the other sentences holds the theme text.

2.3 Layer 2 : Optimization using social spiders

Optimization using social spiders.

Natural Model.

- **Environment**: a set of pickets which serve weaving wire brackets, this pickets have different sizes.
- **Weaving**: weaving is to create a link between the current and the last visited pick
- **Movement**: movement allows the spider to move in the environment on the wire woven by her or by others spider in the same canvas. The selection of the new position dependent upon a finite number of criteria. The wire has a flexibility F which is one of the major criteria of movement of the spider, the flexibility F represents the maximum weight with a correction relative to its diameter that can be held by the wire.
- **Communication**: social spiders communicate with the others in the weaving task, movement or capture prey; communication can be done by two different methods; by vibration on the wire or by the concentration of hormonal substances that spider left on a wire. Each vibration intensity and each concentration of substances has a specific meaning to others spiders, and this means that each spider must have two receivers (vibration and concentration).
- **System dynamics**: It is built on the principle of stigmergy: the behavior of agents have effects on the environment (wire-laying) in return behavior is influenced by the context of the agent (environment).

Artificial Model.

- **Environment :** a picket grid (N * N). N is the square root of the number of phrases after layer 1 (pre-summary and the elimination of similar phrases) where each

pickets is representing a sentence, the pickets have different sizes that representing the score of the phrase. Initially, all the wires are woven so as to have a complete graph;

The number of spiders is equal to or less than the number of phrases, each spider is placed on a pole (phrase) randomly.

- **Weaving :** the wires are woven in the beginning of each iteration in order to have a complete graph. The similarity s_{xy} between two phrases ph_x ph_y represents the diameter of wire woven between the two picket x and y associate to phrases ph_x to and p_{hy}, as given the similarity is commutative (s (x, y) = s (y, x)): the diameter of wire woven between the two picket x and y will be a uniform.
- **Movement:** movement of the spider is incremental and random; Every spider save in its memory every way which she followed. To save result, a weight of path should be: Superior to the "lower threshold of the summary rate" and Less the "upper threshold of the summary of rates."

We associated to the social spider *i* in iteration *j* a P_{ij} weight initialized to zero and it equal to the sum of the weights of *k* SCORE sentences whose social spider *i* have visited during the iteration *j*.

The wire has a flexibility that F depends on its diameter is constant and represents the maximum weight that can load on itself, with artificial model F Is defined as follows.

$$Flexible\ (fil_{ij}) = Seuil\ supérieur\ de\ taux\ de\ résumé\ *\ \ diametre\ (fil_{ij}) (4)$$

$$diametre\ (fil_{ij}) =\ similarité\ (\ phrase_i, phrase_j) (5)$$

Noting that:

- Abstract rate threshold is constant.

If i social spider during operation j with Pij weight passes through the wire (x, y), it will execute this pseudo-algorithm:

```
If Pij is lower than F(x,y) then
  the spider will go to the wire (x,y)
  updating the current path
  Update the weight Pij,
If the wire is torn
```

Social spider i will go into pause waiting for the end of the iteration j.

We will give these two observations:

(a) F(x,y) is higher than F(w,z) is equivalent to say that the similarity between the sentence x and the sentence y is greater than the similarity between the sentence w and the sentence z because “upper threshold of the summary of rates” is constant.

(b) The interpretation of F(x,y) is higher than F(w,z), is that by optimizing with social spider: if choice between wire (x,y) and the wire (w,z) the spider will

choose the first wire because it safe for her . If his current weight Pij is high; the second wire risk to tear.

From observations A and B, we can deduce that the optimization is to minimize the weight of the summary, to maximize the similarity to preserve the theme of the candidate summary, while respecting the dice constrained utility and semantics represented by the interval [lower threshold summary of rates, upper rates higher threshold] noting that the lower and upper thresholds are summarized determined and fixed as language experts.

- ***The utility constraint*:** Automatically produce a summary with higher summary score "upper threshold of the summary of rates," is not helpful.
- ***The semantic constraint:*** Automatically produce a summary with lower summary score "lower threshold of the summary of rates," losing a lot of semantics.

- **End of iteration :** when all the spider will be in pause state, the iteration j will be declared finished, the spider will rewoven the spiders randomly choose their new start position and start the iteration j + 1.
- **Communication :** Each spider leaves a trace on hormonal stakes visited so that other spiders will not take this part of the way. First it ensures diversity between different summaries candidate that is greater coverage suspected combination spiders consider this shift pickets the number they share with each spider that operates on the canvas, and moves with the constraint of not exceeding M common stake in the same order with another spider .

  Secondly, communication is used to avoid the repetition of sentences in the summary. In cases where social spider returns while moving on a picket that it been already have been visited by itself in current iteration it makes a flashback and continues his trip without considering this visit.

  The duration of evaporation of communication hormone spider is equal to an iteration, it should be noted that the hormone density can not be cumulative.
- **System dynamics**: It is built on the principle of stigmergy: the behavior of agents have effects on the environment, in return behavior is influenced by the context of the agent.

  Each spider keeps in mind, the best visited paths, after a number of spider iterations, every spider returns the best paths.

- **Path :** is a series of picket visited in chronological order, and is a summarization. Recall that each picket is a phrases (see the initial state).
- **End of the optimisation of the social spiders:** when the number of iterations performed reached the maximum number of iterations, each spider returns all paths (where each path, is a candidate summary). Was associated with each path or summarization ie a set of candidate evaluations indices. And launching a voting algorithm compared these evaluation indices to choose the best candidate summary to remember.

2.4 Layer 3: evaluation and vote

Candidates generated by the previous layer abstracts will be evaluated by several evaluations metric, and then we will classify pairs. R1 and R2 are two abstracts candidate rate by N metric evaluation, the number of associated point R1 represents the number of evaluation indicating that the quality of R1 is greater than or equal to R2 and aims to it. The summary with most points will win the duel and will face another until there will be more challenger. Summary will be declared the winner as the best back to resume.

3 Experimentation

Under the assumption that the weight of a sentence indicates its importance in the document and under assumption that two similar sentences have the same meaning; we applied the algorithms summarized by extracting in occurrence Scoring and similarity phrases. Our method is oriented for the moment to the generation of a mono-document summary using a biomimetic approach (Social Spider).

3.1 Used corpus

Was used as the text corpus "Hurricane" in French, which contains a title and 20 sentences and 313 words, after the pretreatment process and vectorization bag of words, we get 171 different token. And we have took three references summaries produced successively by Summarizer CORTEX, Essential Summarizer, and a summary produced by a human expert.

3.2 Validation

We evaluated the summaries produced by this algorithm with the metric ROUGE (Lin 2004) which compares a candidate summary (automatically produced) and Summary Reference (created by human experts or other automatic summarization systems known).

The evaluation measure Recall - Oriented Understudy for Gisting Evaluation.

We evaluate the results of this work by the measure called Recall - Oriented Understudy for Gisting Evaluation (ROUGE) proposed by (Lin, 2004) involving the differences between distributions of words.

$$ROUGE\ (N) = \frac{\sum_{s \in R_{ref}} \sum_{s \in R_{can}} \text{Co} - \text{occurences}\ (R_{ref}, R_{can}, N)}{\text{Nbr} - \text{NGramme}\ (N)_{R_{ref}}} (6)$$

F-measure for the evaluation of automatic extraction summaries.

We have proposed in our work before an adaptation of the F-measure for the validation of automatic summarization by extraction, as this technique is based on phrases to keep and delete

Confusion matrix		Candidate sumarry	
Automatic summary		word K	word R
Reference summary	Word K	X	Y
	Word R	Z	W

Word K : number of words to keep
Word R: number of words to remove

Table 1. Adaptation of the F-measure for the validation of automatic summarization

From the confusion matrix, we can calculate: the recall, precision than we combined the two measures to calculate the F-Measure like that:

$$F - Mesure = \frac{2 * (Précision * Rappel)}{(Précision + Rappel)} (7)$$

3.3 Result

Results of layer 1 : before optimisation with social spiders .

Phrases score threshold / Similarity		0,60						0,65					
threshold simila-rity	Metric evaluation	REG	Cortex	Humain	Nbr word	Nbr Phrase	Reduced rates	REG	Cortex	Humain	Nbr word	Nbr Phrase	Reduced rates
0,60	ROUGE	0,67	0.71	0.55	245	15	21,72%	0,67	0,68	0,52	224	13	28,43%
	F-Mesure	0,49	0,46	0,32				0.55	0.50	0.47			
0,65	ROUGE	0,65	0,69	0,55	232	14	25,87%	0,72	0,68	0,53	221	12	29,39%
	F-Mesure	0.51	0.49	0.37				0.58	0.54	0.52			
0,70	ROUGE	0,61	0,68	0,51	230	14	26,51%	**0,71**	**0,69**	**0,56**	**208**	**10**	**33,54%**
	F-Mesure	0.57	0.51	0.44				**0.64**	**0.62**	**0.55**			
Phrases score threshold / Similarity		0,70						0,75					
threshold simila-rity	Metric evaluation	REG	Cortex	Humain	Nbr word	Nbr Phrase	Reduced rates	REG	Cortex	Humain	Nbr word	Nbr Phrase	Reduced rates
0,60	ROUGE	**0,73**	**0,74**	**0,55**	**193**	**10**	**38,33%**	0,67	0,67	0,58	123	6	60,70%
	F-Mesure	**0.61**	**0.64**	**0.59**				0.43	0.47	0.45			
0,65	ROUGE	0,67	0,70	0,57	180	8	42,49%	0,68	0,68	0,58	113	5	63,89%
	F-Mesure	0.58	0.59	0.56				0.42	0.45	0.41			
0,70	ROUGE	0,71	0,68	0,54	143	7	54,31%	0,71	0,74	0,64	110	4	64,85%
	F-Mesure	0.57	0.55	0.55				0.39	0.45	0.38			

Fig. 1. Result of Layer 1 : before optimization with Social Spider

- In Yellow: the local optimal candidate summary before optimization, quoted just for illustration, but will not be used for optimization with social spiders.
- In the Green: abstract global optimal candidate before optimization, which will be used for optimization with social spiders,

Results of layer 3 : after optimization with the Social Spider and VOTE.

We used two social spiders parameter combined with Number of iterations = 500

	Combine1	Combine 2
Threshold higher discount rate	55% =0,55	50%= 0,50
Threshold lower discount rate	27,5%=0.275	30%=0,30
Number of spiders	3	3
Maximum number of common stake in the same order	5	5

	Metric evaluation	REG	Cortex	Humain	Nbr word	Nbr Phrase	Reduced rates	Execution time
Before Optimization	ROUGE	**0,71**	**0,69**	**0,56**	**208**	**10**	**33,54%**	**819 ms**
	F-Mesure	**0.64**	**0.62**	**0.55**				
Optimizing social spider (Combine 1)	ROUGE	0.72	0.73	0.60	205	9	34,50%	3602 ms
	F-Mesure	0.66	0.67	0.57				
Optimizing social spider (Combine 2)	ROUGE	0.72	0.75	0.61	195	9	37,69%	2762 ms
	F-Mesure	0.68	0.72	0.66				

Fig. 2. Optimization *of 1st summary candidate score threshold=0.*65, threshold of similarity=0.70

	Metric evaluation	REG	Cortex	Humain	Nbr word	Nbr Phrase	Reduced rates	Execution time
Before Optimization	ROUGE	**0,73**	**0,74**	**0,55**	**193**	**10**	**38,33%**	**833 ms**
	F-Mesure	**0.61**	**0.64**	**0.59**				
Optimizing social spider (Combine 1)	ROUGE	0.68	0.66	0.47	187	9	40,25%	3859 ms
	F-Mesure	0.64	0.62	0.55				
Optimizing social spider (Combine 2)	ROUGE	0.75	0.78	0.62	190	9	39,29%	2591 ms
	F-Mesure	0.65	0.66	0.6				

Fig. 3. Optimization *of 2nd summary candidate score threshold=0.*70, threshold of similarity=0.60

We conducted a series of experiments to find and fix the most optimal parameters of social spiders.

3.4 Interpetation

We experimented document "Hurricane" using the coding TFC for the first stage (scoring) and several similarity distances (second stage) to try to detect theUSER sensitive about the best results we summarized validated by the metric RED by com-

paring the summary reference from REG system COTREX and a human expert who summed us the text "Hurricaine". All tests on data representation parameters were performed to éviterde misjudge our new approach based on a biomimetic approach in this case social spiders.

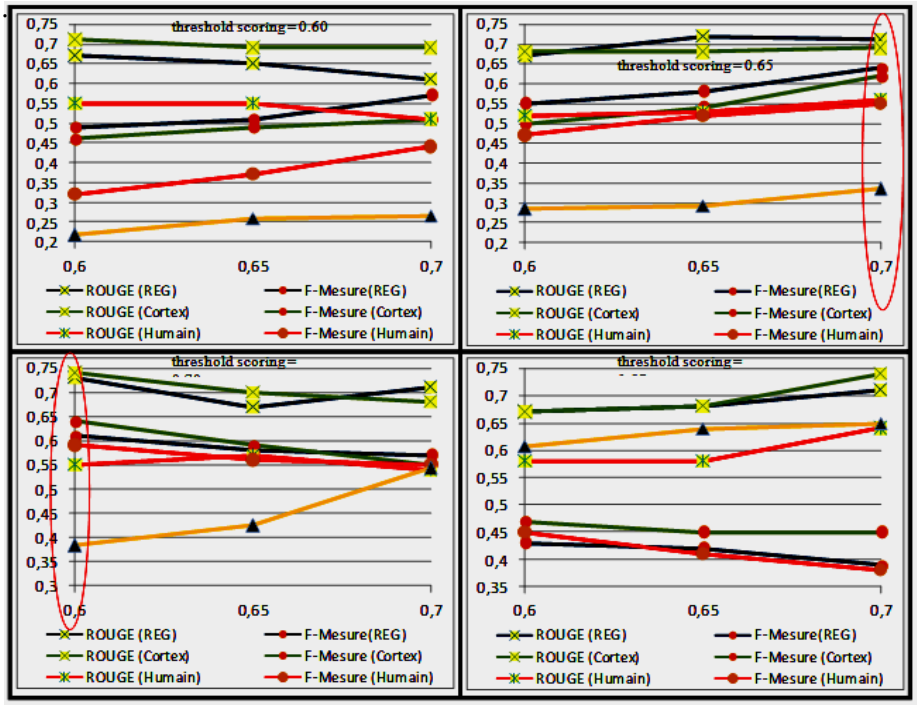

Fig. 4. Summary Evaluation graph before optimization (layer 1)

The first sub-graph (top left corner) indicates incoérence between the two F-evaluation metric measurement and ROUGE incoérence this is resulting from a false assessment of ROUGE summary. This is explained by the weak against the ROUGE summary negligible rate reduction: in fact a summary has low reduction rate will have the lesco-occurrences of N-grams number between him and a set of reference summaries Rref

larger than a summary has greatly reduced rates.

The second sub-graph (top right corner) and the third sub-graph (bottom left) sub-graph shows complete coherence between the three evaluation indexes: reduction rate, F-Measure and ROUGE. While the fourth sub.

According to the experimental set of results when we set the target parameter values, it has turn out that:

(a) Increasing number of iterations and the increase in social spiders influences the execution time, the candidate summary quality is not reached by the change of these two parameters

(b) Maximum number of common stake in the same order minimizes the number of abstracts same candidate before the vote and can cover the maximum possible case

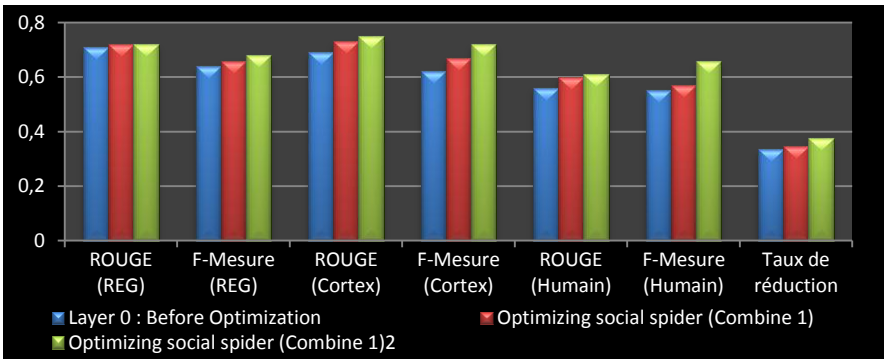

Fig. 5. Optimization of the first summary candidate score threshold=0.65, threshold of similarity= 0.70

The graph below shows explicitly that the second parameter optimization combined with social spiders return results better compared to the first combination, this is explained by the given interval of utility and semantics represented by two thresholds: upper and lower discount rate is reduced, which allows well-directed social spider. While the first combined with a wider interval, that channels less the optimization work.

We note that the execution time optimization combined with the first is greater than the second combines this means that the search field combines 1 is greater than the second combination.

4 Conclusion and perspective

In this article, we presented new ideas: the first is to have used two techniques of extraction summary after another to improve the rate of reduction without loss of semantics.

The second idea is the use of a biomimetic approach that has the representation of strength graph, social spiders can almost total coverage on a graph using the communication module.

Given the results obtained, our approach based on a biomimetic approach (social spiders) can help solve one of the problems of textual data exploration and visualization will.

Prospects we will try to improve this approach using the WordNet thesaurus, and use a summary based on feelings using the SentiWordNet. We'll also try to explore other biomimetic methods. For nature still has not revealed all the secrets.

5 Reference

1. Luhn, H. P. (1958). The automatic creation of literature abstracts. IBM Journal of research and development, 2(2), 159-165.
2. Edmundson, H. P. (1963). Automatic Abstracting, TRW Computer Division, Thompson Ram Wooldridge. Inc., Canoga Park, CA.
3. DeJong, G. (1982). An overview of the FRUMP system. Strategies for natural language processing, 113.
4. Fum, D., Guida, G., & Tasso, C. (1982, July). Forward and backward reasoning in automatic abstracting. In Proceedings of the 9th conference on Computational linguistics-Volume 1 (pp. 83-88). AcademiaPraha.
5. Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1997). Automatic text structuring and summarization. Information Processing& Management, 33(2), 193-207.
6. Mitra, M., Buckley, C., Singhal, A., &Cardie, C. (1997, June). An Analysis of Statistical and Syntactic Phrases. In RIAO (Vol. 97, pp. 200-214).
7. Teufel, S., &Moens, M. (1999). Argumentative classification of extracted sentences as a first step towards flexible abstracting. Advances in automatic text summarization, 155, 171.
8. Mani, I., House, D., Klein, G., Hirschman, L., Firmin, T., &Sundheim, B. (1999, June). The TIPSTER SUMMAC text summarization evaluation. In Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics (pp. 77-85). Association for ComputationalLinguistics.
9. Kim, S. N., Medelyan, O., Kan, M. Y., & Baldwin, T. (2010, July). Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In Proceedings of the 5th International Workshop on Semantic Evaluation (pp. 21-26). Association for Computational Linguistics.
10. Boudin, F., & Morin, E. (2013, June). Keyphrase Extraction for N-best reranking in multi-sentence compression. In North American Chapter of the Association for Computational Linguistics (NAACL).
11. Hovy, E., Lin, C. Y., Zhou, L., & Fukumoto, J. (2006, May). Automated summarization evaluation with basic elements. In Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC 2006) (pp. 604-611).
12. Donaway, R. L., Drummey, K. W., & Mather, L. A. (2000, April). A comparison of rankings produced by summarization evaluation measures. In Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization-Volume 4 (pp. 69-78). Association for Computational Linguistics.
13. Cuevas, E., Cienfuegos, M., Zaldívar, D., & Pérez-Cisneros, M. (2013). A swarm optimization algorithm inspired in the behavior of the social-spider. Expert Systemswith Applications, 40(16), 6374-6384.
14. Hamou, R. M., Amine, A., &Rahmani, M. (2012). A new biomimetic approach based on social spiders for clustering of text. In Software Engineering Research, Management and Applications 2012 (pp. 17-30). Springer Berlin Heidelberg.
15. Hamou, R. M., Amine, A., &Lokbani, A. C. (2012). The Social Spiders in the Clustering of Texts: Towards an Aspect of Visual Classification. International Journal of Artificial Life Research (IJALR), 3(3), 1-14.