

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Term Extraction Using Hybrid Fuzzy Particle Swarm Optimization

Mohammad Syafrullah and Naomie Salim
Faculty of Computing, Universiti Teknologi Malaysia, Skudai,
Johor, 81310, Malaysia

Abstract: Term extraction is one of the layers in the ontology development process, which has the task to extract all the terms contained in the input document automatically. The objective of this process is to generate a list of terms that are relevant to the domain of the input document. In this study, we present a hybrid method for improving the precision term extraction using a combination of Continuous Particle Swarm Optimization (PSO) and Discrete Binary PSO (BPSO) to optimize fuzzy systems. In this method, PSO and BPSO are used to optimize the membership functions and rule sets of fuzzy systems, respectively. The method was applied to the domain of tourism documents and compared with other term extraction methods: TFIDF, Weirdness, GlossaryExtraction and TermExtractor. From the experiment conducted, the combination of PSO-BPSO showed their capability to generate an optimal set of parameters for the fuzzy membership functions and rule sets automatic adjustment. Findings also showed that fuzzy system performance after optimization achieved significant improvements compared with that before optimization and gave better results compared to those four algorithms.

Key words: Term extraction, particle swarm optimization, fuzzy system, text mining

INTRODUCTION

Term extraction is a prerequisite for all aspects of ontology learning. A term is an instance of a recognisable entity in a multimedia corpus that conveys a single meaning within a domain (concept). A recognizable entity is something that can be recognized in corpora, such as words or phrases in textual corpora, or areas in images. Since terms “lexicalize” a concept, terms found in a corpus usually represent candidate concepts that can enrich an ontology. Thus, the main objective of term extraction is the identification of terms in a corpus that possibly convey concepts, which can be used for enriching an ontology.

Many methods have been presented in the literature, mainly for the processing of textual corpora. Most of these are based on linguistic method, terminology and NLP method, while others are based on a statistical/information retrieval method (Cimiano, 2006). Many linguistic methods use shallow text processing techniques such as tokenizer, Part-Of-Speech (POS) tagger and syntactic analyzer (parser). Text-to-Onto, for example, use a linguistic method called Saarbrücken Message Extraction System (SMES) in their system architecture to produce a list of terms from the input text (Maedche and Staab, 2000). Another system, SVETLAN, uses the syntactic analyzer, Sylex to find list of terms from the input text (De Chalendar and Grau, 2000).

Statistical methods usually try to determine the significance of each word with respect to other words in a corpus based on word occurrence frequencies. Various works have been applied in statistical methods for term extraction. Most of them are based on an information retrieval method for term indexing (Salton and Buckley, 1998; Yates and Neto, 1999; Witten *et al.*, 1999; Medelyan and Witten, 2005). Other methods use the notion of “weirdness” (Ahmad *et al.*, 1999), statistical measurement of frequency occurrence of multi-word terms (Frantzi *et al.*, 2000), mutual information and Log Likelihood Ratio (Panel and Lin, 2001), domain specificity (Park *et al.*, 2002; Kozakov *et al.*, 2004), latent semantic indexing (Fortuna *et al.*, 2005) and domain pertinence (Navigli and Velardi, 2004; Sclano and Velardi, 2007).

Clustering techniques also play an important role in term extraction: recognizable entities can be clustered into groups based on various similarity measures with each cluster being a possible term (comprising synonyms). (Kietz *et al.*, 2000; Agirre *et al.*, 2000; Faatz and Steinmetz, 2002) employed clustering techniques and other resources such as www and WordNet to successfully extract terms. Additionally, both frequency and clustering based approaches can be substantially enhanced through the use of natural language processing techniques (such as morphological analysis, part-of-speech tagging and syntactic analysis), as terms are usually noun phrases or

often obey specific part-of-speech patterns (Haase and Stojanovic, 2005). Finally, morphological clues (such as prefixes and suffixes) can be extremely useful for some domains: suffixes such as “-fil” and “-it is” quite often mark terms in medical domains (Haase and Volker, 2008).

Esmin and Lambert-Torres (2006, 2007, 2010) and Lambert-Torres *et al.* (2000) have shown an efficient PSO based approach to construct and optimize a fuzzy rules base and fuzzy membership function. This method assumes that the fuzzy logics can also be formulated as a space problem, where each point of fuzzy sets corresponds to a fuzzy logic i.e. represents membership functions and rule base. Another work by Kiani and Akbarzadeh (2006) proposed a technique for combining a Genetic Algorithm (GA) and Genetic Programming (GP) to optimize rule sets and membership function of fuzzy systems.

As term extraction is an important task not only for concept discovery for ontology learning, but also for textual information extraction and retrieval, this study proposes a term extraction method using a hybrid method of Continuous Particle Swarm Optimization (PSO) and Discrete Binary PSO (BPSO). This method was used to optimize and automatically adjust the membership functions and rule sets of fuzzy systems, respectively.

MATERIALS AND METHODS

This section presents a new approach to extract terms based on an optimized fuzzy inference system, which takes into account several kinds of features including domain relevance, domain consensus, term cohesion, first occurrence and length of noun phrase.

Initially, syntactic parser performs a syntactic analysis on every input sentence from the input document and extracts a list of syntactic information (Noun Phrase-NP). All stop words were removed from each of the lists of NP. Finally, the list of NP was stemmed to produce a list of clean NP, as the term candidate. Each term candidate is associated with a vector that contains five features.

The features extracted in the previous step were used as inputs to the fuzzy inference system. In this approach, we want to maximize the accuracy of the fuzzy system output while optimizing its running time, by optimizing the membership function and the rule sets. Therefore, a hybrid of PSO and BPSO was used to find the optimal rule-based system.

Fuzzy system: Fuzzy systems are used in a wide number of applications areas, especially on fuzzy control problems. These systems can be considered as knowledge-based systems, incorporating human

knowledge into their knowledge base through a fuzzy inference system and fuzzy membership functions. The definition of these fuzzy inference systems and fuzzy membership functions is generally affected by subjective decisions, having a great influence over the performance of the system.

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. The fuzzy inference process comprises five parts: Fuzzification of the input variables, application of the fuzzy operator (AND or OR) in the antecedent, implication from the antecedent to the consequent, aggregation of the consequents across the rule sets and defuzzification.

Fuzzy rules are linguistic IF-THEN constructions that have the general form "IF A THEN B", where A and B are collections of propositions containing linguistic variables. A is called the premise and B is the consequence of the rule. The fuzzy rules are written as:

- IF x is A THEN y is B, where x and y are linguistic variables; and A and B are linguistic values determined by fuzzy sets on the universe of discourses X and Y, respectively (Fig. 1)

Continuous PSO: The particle swarm optimization (PSO) algorithm (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995) has been introduced as an optimization technique for use in real-number spaces. It is originally designed for continuous optimization problems.

PSO is inspired by this kind of social optimization, where a problem is given and some way to evaluate a proposed solution exists in the form of a fitness function. A communication structure or social network is also defined, assigning neighbours for each individual to interact with. Then, a population of individuals, defined as random guesses at the problem solutions is initialized. These individuals are candidate solutions. They are also known as particles, hence the name particle swarm. In PSO, each particle in a population has a position and a velocity vector. Derived from utilization of other particles, individual best's and global best's positions which will be clarified later, a velocity vector is generated. That particle then modifies its own position due to the velocity vector for which the achievement of coordination and communication with the experience of other individuals is not ignored.

A PSO includes three basic steps: Evaluation of the fitness values of each particle, updating the values and positions of individual and global best and updating the velocities and positions of each individual in the population according to the Eq. 1 and 2:

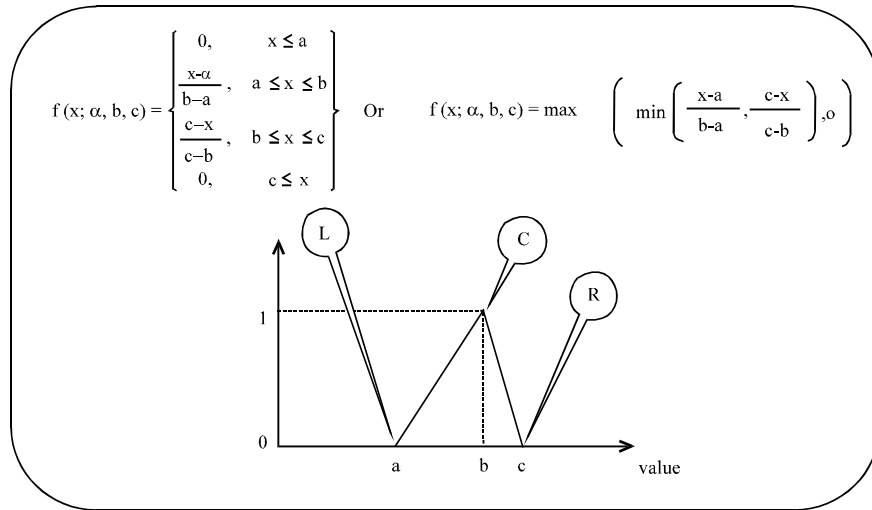


Fig. 1: Triangular membership functions

```

Randomly generate an initial population
repeat
  for i = 1 to population_size do
    if f(x̄i) < f(p̄i) then p̄i = x̄i
    p̄g = min(p̄neighbours)
    for d = 1 to dimensions do
      velocity_update()
      position_update()
    end
  end
until termination criterion is met
    
```

Fig. 2: PSO algorithms

$$\vec{v}_i \leftarrow w\vec{v}_i + c_1\vec{r}_1 \odot (\vec{p}_i - \vec{x}_i) + c_2\vec{r}_2 \odot (\vec{p}_g - \vec{x}_i) \quad (1)$$

$$\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i \quad (2)$$

where, \vec{x}_i is the current position of the i -th particle in the swarm, \vec{v}_i is the velocity of the i -th particle, \vec{p}_i is the best position found by the i -th particle so far (local best). The best position found from the particle's neighbourhood (global best) is then stored in a vector \vec{p}_g . \vec{r}_1 and \vec{r}_2 are the numbers uniformly chosen from $[0, 1]$. c_1 is a positive constant called the coefficient of the self-recognition component and c_2 is a positive constant called the coefficient of the social component. The variable w is called the inertia factor, whose value is typically set up to vary linearly from 1 to 0 during the search process (Fig. 2).

Discrete binary PSO: Continuous PSO is mainly applied in the area of continuous/real number space and rarely in

discrete space. However, many optimization problems are set in a space featuring discrete binary or qualitative distinctions between variables, for example routing problem, scheduling/timetabling problem and data mining. To meet the need, Kennedy and Eberhart (1997) proposed a discrete binary version of PSO for binary problems (Eberhart and Shi, 1998). In their model, a particle will decide on "yes" or "no", "true" or "false" etc. This binary value can also be a representation of a real value in binary search space.

Discrete PSO essentially differs from the original (or continuous) PSO in two characteristics. First, the particle is composed of the binary variable. Second, the velocity must be transformed into the change of probability, which is the chance of the binary variable taking the value one. In the binary PSO, the particle's personal best and global best is updated as in continuous version. The major difference between binary PSO and continuous version is that velocities of the particles will change to one of the two decisions $[0 \text{ or } 1]$. A sigmoid transformation as in (Eq. 3) is applied to make the velocities value into a range of $[0 \text{ or } 1]$ and force the component values of the particle's position to be 0 or 1. The equation for updating the particle's position is shown in Eq. 4:

$$\text{Sigmoid}(\vec{v}_i) = \frac{1}{1 + e^{-\vec{v}_i}} \quad (3)$$

$$\vec{x}_i = \begin{cases} 1, & \text{if rand() < sigmoid}(\vec{v}_i) \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Feature definition: All term candidates and their features were extracted from text. These entire term candidates

were selected from noun phrases. First of all noun phrases were extracted with the syntactic parser and all stop words were removed from candidates. Five features (f1, f2, f3, f4, f5) were chosen to characterize the noun phrases in the documents. The features used are domain relevance, domain consensus, term cohesion, first occurrence and length of noun phrase.

Domain relevance (f1): Domain relevance can be given according to the amount of information captured in the target document with respect to contrastive documents. Let D_k as the domain of interest (a set of relevant documents) and $\{D_1, \dots, D_n\}$ as sets of documents in another domain, domain relevance of a term t in class D_k be computed as (Navigli and Velardi, 2004; Sclano and Velardi, 2007) Eq. 5 and 6:

$$DR(t, D_k) = \frac{P(t|D_k)}{\max_{1 \leq j \leq n} (P(t|D_j))} \quad (5)$$

$$E(P(t|D_k)) = \frac{f_{t,k}}{\sum_{t \in D_k} f_{t,k}} \quad (6)$$

where, $f_{t,k}$ is the frequency of term t in the domain D_k .

Domain consensus (f2): Domain consensus measures the distributed use of a term in domain D_k . Domain consensus is expressed in Eq. 7 and 8 as follows (Navigli and Velardi, 2004; Sclano and Velardi, 2007):

$$DC(t, D_k) = \sum_{d \in D_k} \left(P(t|d) \cdot \log_2 \left(\frac{1}{P(t|d)} \right) \right) \quad (7)$$

$$E(P(t|d_j)) = \frac{f_{t,j}}{\sum_{d \in D_k} f_{t,j}} \quad (8)$$

Term cohesion (f3): Term cohesion is used to calculate the cohesion of the multi-word terms. The measure is proportional to the co-occurrence frequency and the length of the term (Eq. 9) (Park *et al.*, 2002; Kozakov *et al.*, 2004):

$$TC(t) = \frac{|t| \cdot \log_{10}(f(t)) \cdot f(t)}{\sum_{w \in t} f(w_i)} \quad (9)$$

where, w_j are the words composing the term t .

First occurrence (f4): First occurrence of a term is calculated as the distance of a phrase from the beginning of a document in words, normalized by the total number of words in the document. The result represents the

proportion of the document preceding the phrase's first appearance (Eq. 10) (Witten *et al.*, 1999; Medelyan and Witten, 2005):

$$FO(t, d) = \frac{\sum_{(w_j \in d | 1 \leq w \leq t)} f(w_i)}{\sum_{\{w_i \in d\}} f(w_i)} \quad (10)$$

Length of noun phrase (f5): Length of noun phrase score is calculated as frequency times (Eq. 11) length (in words) (Barker and Cormacchia, 2000):

$$LNP(t) = f(t) \cdot |t| \quad (11)$$

Fuzzy PSO model: Several assumptions are used for the model formulation:

- Triangular membership functions were used for both input and output variables
- Complete rule-base was considered. A rule-base is considered complete when all possible combinations of input membership functions of all the input variables participate in fuzzy rule-base formation

The features extracted in the previous section are used as inputs to the fuzzy inference system. We use triangular membership functions which are specified by three parameters (a, b, c) as in Fig. 1. Parameters a and c locate the "feet" of the triangle and parameter b locates the peak of the triangle.

The input membership function for each feature is divided into three fuzzy sets which are {low, average, high}. For instance, membership functions of the first occurrence are shown in Fig. 3.

The output membership functions are shown in Fig. 4, which is divided into 3 fuzzy sets: {unimportant, average, important}.

The most important part in this procedure is the definition of fuzzy IF THEN rules. We define full rule sets (243 rules). For example, our rules are demonstrated as follows (Fig. 5):

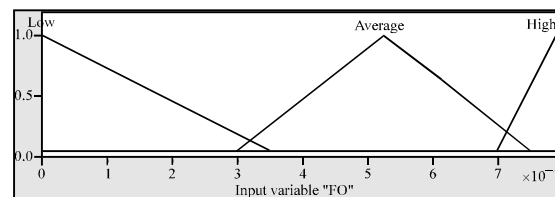


Fig. 3: Membership function of first occurrence

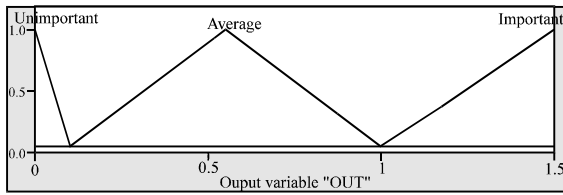


Fig. 4: Membership function of output

IF (DR is low) and (DC is average) and (TC is low) and (FO is low) and (LK is high) then (OUT is average)

Fig. 5: Samples of IF-THEN rules

Table 1: Parameter setting of particle swarm based optimization

Parameters	Value
No. of particles	40
Maximum No. of iterations	500
c1	2
c2	2
W	(0.5+(random/2))

Optimization by PSO: The objective of this optimization process is to find the optimal adjustment values of membership function. For each fuzzy membership function, there are three parameters as shown in Fig. 1: L (left), C (centre), R (right).

The integration of PSO algorithms with a fuzzy system was made as follows:

- The parameters are the adjustment factor centres (k) and widths (w) of each fuzzy set. Parameter k makes each membership function move to the right or left with no change in the form. The membership function shrinks or expands through parameter w
- These parameters act as particles and look for the global best fitness
- The initial set of particle swarm optimization variables is shown in Table 1
- After the parameters had been adjusted using optimization method, this parameter was used to evaluate the performance of the fuzzy system using fitness function as in Eq. 13
- For the adjustment of membership functions, the following equations were defined (Eq. 12):

$$\begin{aligned}
 L &= (L+k)-w \\
 C &= (C+k) \\
 R &= (R+k)+w
 \end{aligned}
 \tag{12}$$

- The process is repeated until the termination criterion is met (Fig. 6):

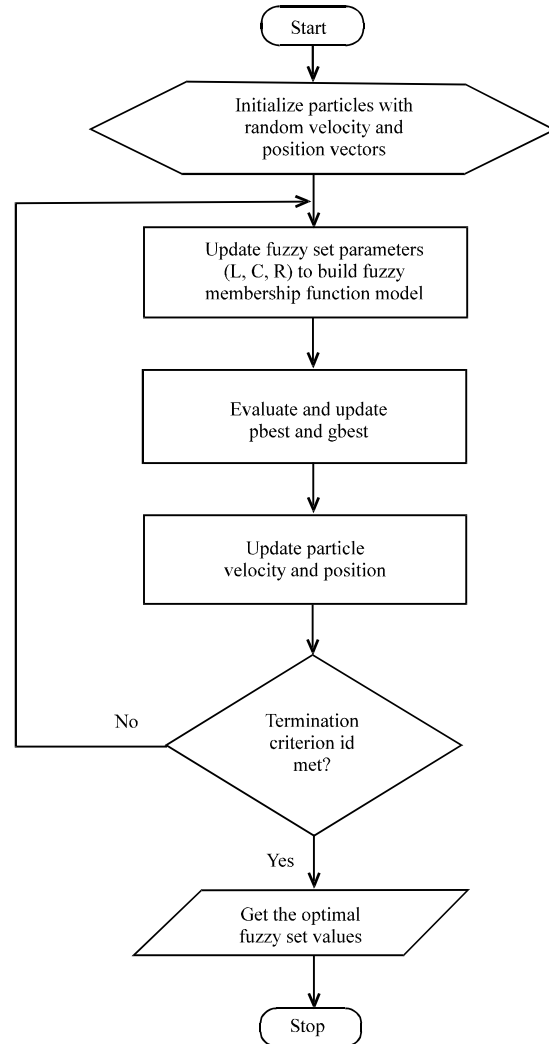


Fig. 6: Flowchart of fuzzy membership function adjustment using PSO

$$\text{Fitness} = \max_{1 \leq n \leq m} (\text{FIS out}_n) \cdot \max \left(\sum_{i=1}^{|\text{extracted}|} |t_i \in \text{goldstandard}| \right) \tag{13}$$

Optimization by BPSO: In this research, we used BPSO to optimize the IF-THEN rules of the fuzzy system. The integration of BPSO algorithms with the fuzzy system was made as follows:

- The parameters are the rule number of rule sets, which will determine whether the rule is "select" or "not select" in the rule sets
- These parameters act as particles and look for the global best fitness
- The initial set of particle swarm optimization variables is shown in Table 1

- After the parameters had been adjusted using the optimization method, the fitness of rules was obtained by bonding the rules with membership functions. The fitness was evaluated using Eq. (14)
- The process is repeated until the termination criterion is met:

$$\text{Fitness} = \max_{1 \leq n \leq m} (\text{FIS out}_n) \cdot \min(\text{rule}) \cdot \max \left(\sum_{i=1}^{|\text{extracted}|} |t_i \in \text{goldstandard}| \right) \quad (14)$$

Datasets

Tourism: The data used for testing and evaluating the proposed method was document collection about tourism [http://olc.ijs.si/lpTxt/] and has already been used by Cimiano (2006). The collection with approximately a million tokens size consists of 1800 text documents, describing countries, cities, places, regions, sights etc. from all continents. In this experiment, we used 300 documents as datasets.

Reuters-21578: The documents in the Reuters-21578 collection appeared on the Reuters newswire in 1987. We converted all the documents into 22 plain text files (reut2-000.txt until reut2-021.txt); the size of the text corpus is about 218 Mio tokens. Reuters-21578 was used in the experiment as contrastive documents.

RESULTS AND DISCUSSION

After execution of the optimization process, we can see that PSO has successfully adjusted fuzzy membership function and fuzzy rule sets. The fuzzy set for every membership function has been adjusted from standard fuzzy membership function and the number of rule sets is reduced from 243 rules to 190 rules (Fig. 7 and 8).

In this extraction stage, we evaluate the precision of our proposed method at seven points: top 25, 125, 225, 325, 425, 525 and 625 terms, both for each feature and the scores of the term candidates using the following (Eq. 15):

$$\text{Precision} = \frac{\sum_{i=1}^{|\text{extracted}|} |t_i \in \text{goldstandard}|}{|\text{extracted}|} \quad (15)$$

Table 2: Term extraction precision for each feature

Precision (feature)	No. of terms						
	25	125	225	325	425	525	625
f1	0.48000	0.30400	0.27556	0.22154	0.18353	0.15238	0.13440
f2	0.56000	0.32800	0.31111	0.27077	0.24000	0.21143	0.19200
f3	0.16000	0.10400	0.07556	0.07385	0.08000	0.08381	0.08000
f4	0.20000	0.06400	0.04444	0.05231	0.05882	0.05905	0.05760
f5	0.56000	0.32800	0.31111	0.28308	0.24941	0.21524	0.19360

We compared the terms extracted by the system with the gold standard. Table 2 shows the term extraction precision for each feature for different numbers of terms evaluated.

In order to see the improvement in our proposed method, it was compared with a fuzzy system without using optimization. From the experimental results, it can be seen that our proposed method, combination of PSO and fuzzy system, can increase the precision of the extracted term compared to a fuzzy system without using optimization.

Despite the improvement in accuracy results that were achieved using the Fuzzy PSO model, it was necessary to compare the prediction accuracy results to other algorithms. This study compared the precision of our proposed method with other known algorithms as shown in Fig. 9. It was observed that the precision of the Fuzzy PSO model is much higher than the accuracy of any other algorithms. Table 3 and Fig. 9 show the comparison of the precision between the fuzzy swarm model, fuzzy model and the four other algorithms (TFIDF, Weiridness, GlossaryExtraction and TermExtractor).

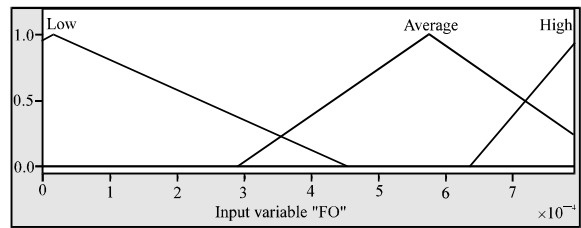


Fig. 7: First occurrence membership function after applying PSO

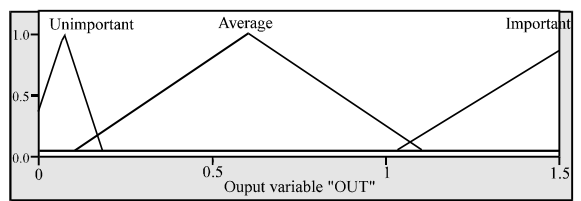


Fig. 8: Output membership function after applying PSO

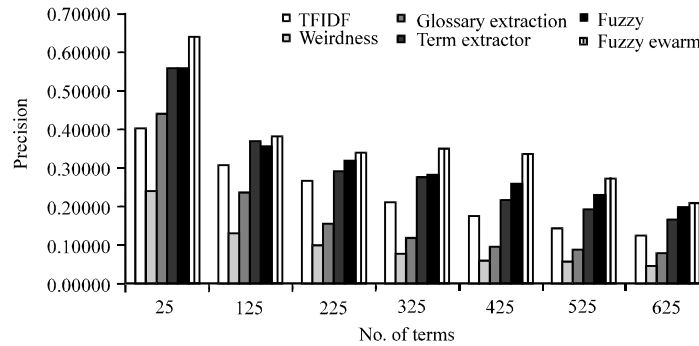


Fig. 9: Comparison of term extraction precision (Fuzzy swarm model, fuzzy, TFIDF, weirdness, glossary extraction and term extractor)

Table 3: Comparison of term extraction precision using different methods

Features	No. of terms						
	25	125	225	325	425	525	625
TFIDF	0.40000	0.30400	0.26667	0.20615	0.16941	0.14095	0.12160
Weiridness	0.24000	0.12800	0.09778	0.07385	0.05882	0.05333	0.04800
Glossary extraction	0.44000	0.23200	0.15111	0.11692	0.09412	0.08571	0.07680
Term extractor	0.56000	0.36800	0.28889	0.27077	0.21412	0.19048	0.16480
Fuzzy	0.56000	0.35200	0.32000	0.28308	0.26118	0.22857	0.19680
Fuzzy swarm	0.64000	0.37600	0.34222	0.35077	0.33412	0.27429	0.20800

Finally, the experimented results conclude that Fuzzy PSO leads to improving the precision results while adjusting fuzzy membership function and keeping the number of fuzzy rules to a minimum with a high degree of accuracy.

CONCLUSION

In this study, a new approach is proposed to extract terms based on an optimizing fuzzy inference system. The continuous PSO is used to optimize the membership functions and Discrete Binary PSO is used to optimize the fuzzy rule sets. Based on the results of the experiment, the PSO has successfully adjusted fuzzy membership function and fuzzy rule sets and improved the performance result in terms of precision. Finally, the proposed method is compared with four other known algorithms (TFIDF, Weiridness, GlossaryExtraction and TermExtractor). Our experimental results show that the use of particle swarm optimization to optimize fuzzy membership function and fuzzy rule sets can significantly improve the precision of the extracted term.

ACKNOWLEDGMENT

This study is funding by Research Management Centre, Universiti Teknologi Malaysia.

REFERENCES

Agirre, E., O. Ansa, E.H. Hovy and D. Martinez, 2000. Enriching very large ontologies using the WWW. Proceedings of the 1st Workshop on Ontology Learning, August 25, 2000, Berlin, Germany, pp: 1-6.

Ahmad, K., L. Gillam and L. Tostevin, 1999. University of surrey participation in TREC 8: Weiridness Indexing for Logical Document Extrapolation and Retrieval (WILDER). Proceeding of the 8th Text Retrieval Conference, November 17-19, 1999, Department of Commerce, National Institute of Standards and Technology, USA., pp: 717-724.

Barker, K. and N. Cormacchia, 2000. Using noun phrase heads to extract document Keyphrases. Proceedings of the 13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, May 14-17, 2000, Canada, pp: 40-52.

Cimiano, P., 2006. Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. 1st Edn., Springer, Germany.

De Chalendar, G. and B. Grau, 2000. SVETLAN' or how to classify words using their context. Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management Methods, Models and Tools, October 2-6, 2000, Juan-les-Pins, France, pp: 203-216.

- Eberhart, R.C. and J. Kennedy, 1995. A new optimizer using particle swarm theory. Proceedings of the 6th International Symposium on Micro Machine and Human Science, October 4-6, 1995, Nagoya, Japan, pp: 39-43.
- Eberhart, R.C. and Y. Shi, 1998. Comparison between genetic algorithms and particle swarm optimization. Proceedings of the 7th International Conference on Evolutionary Programming VII, March 25-27, 1998, Germany, pp: 611-616.
- Esmín, A.A.A. and G. Lambert-Torres, 2006. Fitting fuzzy membership functions using hybrid particle swarm optimization. Proceedings of the IEEE International Conference on Fuzzy Systems, July 16-21, 2006, Vancouver, Canada, pp: 2112-2119.
- Esmín, A.A.A. and G. Lambert-Torres, 2007. Evolutionary computation based fuzzy membership functions optimization. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, October 7-10, 2007, Montreal, Canada, pp: 823-828.
- Esmín, A.A.A. and G. Lambert-Torres, 2010. Generate and optimize fuzzy rules using the Particle Swarm Algorithm. Proceedings of the IEEE International Conference on Systems Man and Cybernetics, October 10-13, 2010, Istanbul, Turkey, pp: 4244-4250.
- Faatz, A. and R. Steinmetz, 2002. Ontology enrichment with texts from the WWW. Proceedings of the 2nd Semantic Web Mining Workshop, August 20, 2002, Helsinki, Finland, pp: 1-15.
- Fortuna, B., M. Grobelnik and D. Mladenic, 2005. Visualization of text document corpus. *Informatica*, 29: 497-502.
- Frantzi, K., S. Ananiadou and H. Mima, 2000. Automatic recognition of multi-word terms: The C-value/NC-value method. *Int. J. Digit. Lib.*, 3: 115-130.
- Haase, P. and L. Stojanovic, 2005. Consistent evolution of OWL ontologies. Proceedings of the 2nd European Semantic Web Conference, May 29-June 1, 2005, Heraklion, Greece, pp: 182-197.
- Haase, P. and J. Volker, 2008. Ontology Learning and Reasoning-Dealing with Uncertainty and Inconsistency. In: *Uncertainty Reasoning for the Semantic Web I*, Da Costa, P.C.G., C. d'Amato, N. Fanizzi, K.B. Laskey and K.J. Laskey *et al.* (Eds.). Springer, USA., pp: 366-384.
- Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. Proceedings of the International Conference on Neural Networks, Volume 4, November 27-December 1, 1995, Perth, WA., USA., pp: 1942-1948.
- Kennedy, J. and R.C. Eberhart, 1997. A discrete binary version of the particle swarm algorithm. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Computational Cybernetics and Simulation, Volume 5, October 12-15, 1997, Orlando, FL., USA., pp: 4104-4108.
- Kiani, A. and M.R. Akbarzadeh, 2006. Automatic text summarization using hybrid fuzzy GA-GP. Proceedings of the IEEE International Conference on Fuzzy Systems, July 16-21, 2006, Vancouver, BC., Canada, pp: 977-983.
- Kietz, J.U., A. Madche and R. Volz, 2000. A method for semi-automatic ontology acquisition from a corporate intranet. Proceedings of the Workshop on Ontologies and Texts, October 2-6, 2000, Juan-Les-Pins, France, pp: 1-14.
- Kozakov, L., Y. Park, T. Fin, Y. Drissi, Y. Doganata and T. Cofino, 2004. Glossary extraction and utilization in the information search and delivery system for IBM technical support. *IBM Syst. J.*, 43: 546-563.
- Lambert-Torres, G., M.A. Carvalho, L.E.B. da Silva and J.O.P. Pinto, 2000. Fitting fuzzy membership functions using genetic algorithms. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Volume 1, October 8-11, 2000, Nashville, USA., pp: 387-392.
- Maedche, A. and S. Staab, 2000. Mining ontologies from text. Proceedings of the 12th International Conference on Knowledge Acquisition, Modeling and Management, October 2-6, 2000, Juan-les-Pins, France, pp: 189-202.
- Maedche, A. and S. Staab, 2003. Mining Ontologies from Text. In: *Knowledge Engineering and Knowledge Management Methods, Models and Tools*, Maedche, A. and S. Staab (Eds.). Springer, Germany, ISBN-13: 9783540411192, pp: 169-189.
- Medelyan, O. and I.H. Witten, 2005. Thesaurus-based index term extraction for agricultural documents. Proceeding of the Joint Conference on European Federation for Information Technology in Agriculture and World Congress on Computers in Agriculture, July 25-28, 2005, Vila Real, Portugal, pp: 1122-1129.
- Navigli, R. and P. Velardi, 2004. Learning domain ontologies from document warehouses and dedicated web sites. *Comput. Linguist.*, 30: 151-170.
- Panel, P. and D. Lin, 2001. A statistical corpus-based term extractor. Proceedings of the 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, June 7-9, 2001, Ottawa, Canada, pp: 36-46.

- Park, Y., R.J. Byrd and B.K. Boguraev, 2002. Automatic glossary extraction: Beyond terminology identification. Proceedings of the 19th International Conference on Computational Linguistics, August 26-30, 2002, Taipei, Taiwan, pp: 772-778.
- Salton, G. and C. Buckley, 1998. Term-weighting approaches in automatic text retrieval. *Inform. Process. Manage.*, 24: 513-523.
- Sclano, F. and P. Velardi, 2007. Termextractor: A web application to learn the shared terminology of emergent web communities. Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications, March 28-30, 2007, Portugal, pp: 287-298.
- Witten, I.H., G.W. Paynter, E. Frank, C. Gutwin and C.G. Nevill-Manning, 1999. KEA: Practical automatic keyphrase extraction. Proceedings of the 4th ACM Conference on Digital Libraries, August 11-14, 1999, California, USA., pp: 254-256.
- Yates, B. and R. Neto, 1999. *Modern Information Retrieval*. 1st Edn., Addison-Wesley, USA.