Cooperation of Evolutionary and Statistical PoS-tagging

Rana Forsati
NLP Research Lab.
Faculty of Electrical and Computer Engineering, Shahid Beheshti University, G. C.,
Tehran, Iran
r_forsati@sbu.ac.ir

Mehrnoush Shamsfard
NLP Research Lab.
Faculty of Electrical and Computer Engineering, Shahid Beheshti University, G. C.,
Tehran, Iran
m-shams@sbu.ac.ir

Abstract— **Part-of-Speech tagging which refers to assignment of syntactic categories to words is a fundamental task in Natural Language Processing (NLP). This paper presents a novel algorithm based on Bee Colony Optimization (BCO) for POS tagging. Experimental results indicate that the proposed algorithm outperforms the other evolutionary-based and tested classical Part-of-Speech -tagging approaches in terms of average accuracy.**

Keywords- bee colony optimization, natural language processing, part of speech tagging

I. INTRODUCTION

Part-of-Speech (PoS) tagging is an important fundamental process in natural language processing (NLP), in which PoS-tags that carry the basic syntactic features of individual words are extracted [1]. Even though several different models and methods have been used for tagging in many languages so far, developing high-quality tagging systems is still a challenging problem. PoS-tagging is the process of assigning the most likely sequence of syntactic categories to every word in a sentence according to its context.

In the last decades, statistical methods have been applied successfully in PoS-tagging [2]. Statistical taggers are designed to identify the most probable tag sequence based on the statistical occurrence of the tag N-gram and word-tag frequencies. Research on these methods has grown a lot in the recent years, probably due to simplicity, language independence [19] and increased availability of large tagged corpora [4]. Comparisons of approaches [10, 17] have shown that in most cases [8] statistical methods yield better results than the other taggers. These methods amounts to maximize a global measure of the probability of the set of contexts (a tag and its neighboring tags) corresponding to a given tagging of the sentence [21]. Subsequently, we require a method to carry out the search in the tagging space which optimizes this measure of probability.

Evolutionary algorithms (EAs), which are stochastic methods based on a search model, define a global function and attempt to optimize its value by traversing the search space. Therefore, evolutionary methods can be used to perform the search of the tagging which optimizes this measure of probability. Results of [21] indicate that the evolutionary approaches for tagging natural language texts achieve more accuracy compared to other statistical approaches.

In the evolutionary-based taggers, each individual is a sequence of tags assigned to the words in a sentence. However, in statistical methods, disambiguation is usually introduced by assigning different probabilities to a given tag, depending on the neighboring tags on both sides of the word [20]. Evolutionary-based taggers are usually ready to provide the solution after obtaining the necessary parameters, such as word-tag and N-gram frequencies. Since stochastic optimization approaches are suitable for avoiding convergence to a local optimal solution, these approaches can be used to find a global optimal solution.

Metaheuristic algorithms such as Simulated Annealing (SA) and Genetic Algorithms (GA) have been previously employed to solve the problem of PoS-tagging [20, 21]. Some attempts related to the use of EAs hybrid systems for PoS tagging have been described in the literature. In [16] a set of inductive logic programs (ILPs) written in Prolog are subjected to evolutionary processes with a suitable crossover operator and mutation replaced by an inductive logic algorithm. Curran and Wong [18] have actually suggested the use of evolved transformations in the Brill Tagger.

In this paper, by modeling PoS-tagging as an optimization problem, we investigate the bee colony optimization algorithms in the tagging problem with the aim of studying the possible relationship between the tags in a sentence. The cooperation between evolutionary algorithms and statistical PoS-tagging is introduced in this paper. By this cooperation the basic elements of the evolutionary algorithms, such as the fitness function is highly simplified by resorting to the statistical PoS-tagging models.

978-1-4673-1479-4/12/$31.00 ©2012 IEEE

The Bee Colony Optimization (BCO) algorithm belongs to the class of stochastic swarm meta-heuristic optimization methods [15]. The BCO is a "bottom-up" approach to modeling where special types of artificial agents are created by analogy of bees. Artificial bees represent agents, which collaboratively solve complex combinatorial optimization problems. Since its inception, BCO has been successfully applied to a wide variety of engineering and management problems [14]. In fact, in optimization problems, we want to search the solution space and in BCO it can be done more efficiently.

Statistical measurements, which are extracted by the statistical taggers, are appropriate alternatives for use as the function to compare the solutions which are extracted by the EAs. The structure of the individual in the proposed algorithm is simply a tag with different probabilities of associated contexts attached to it. Individuals' structure amounts to a tag assignment mechanism that is statistical. The fitness function for the implementation is rather complex. By using the statistical fitness function, we attempt to search for the most probable tag for a word expressed as a component of the solutions. To demonstrate the effectiveness of an algorithm, we have applied the proposed algorithm on a standard corpus and have obtained very good results compared to the other algorithms. The remainder of this paper is organized as follows: Section 2 provides a detailed description of evolutionary-based PoS-tagging algorithms. Section 3 presents the test bed of our experiments and the performance evaluation of the proposed algorithms compared to other algorithms. Finally, section 4 summarizes the conclusions of this work.

II. BEETAGGER: BEE COLONY BASED TAGGER ALGORITHM

In this section, we propose a BCO based algorithm tailored for the problem of PoS-tagging. It is called BEETAGger and aims at labeling each word in the sentence with its syntactic category. In order to tag the sentences using bee colony algorithm, we must first model PoS-tagging as an optimization problem that locates the optimal sequence of tags, with tagging quality as the objective. In the proposed scheme, each word represents one component of a multi-component sentence and each possible solution is a vector of tag values. The following subsections describe BEETAGger algorithm. The algorithm consists of an initialization and many forward and backward passes which are adapted according to the requirements of PoS-tagging problem.

Representation of solutions: The first question to solve tagging problem by BCO is how to represent solutions. Let us consider a sentence S formed by n words $\{w_i, i= 1, 2, \ldots, n\}$ in which each w_i (i^{th} word) has k_i possible tags ($1 \le i \le n$). To represent this situation we decompose PoS-tagging problem of a given sentence S into n stages where each stage represents one word of a sentence, i.e. the first stage represents the first word in the sentence; the second stage represents the second word in the sentence and so on. The nodes in each stage show the valid tags for the corresponding word which are taken from a dictionary. Therefore, in each stage, the number of nodes is equal to the number of possible tags for that word. For unknown words that are neither in the lexicon nor in the training data, all possible PoS-tags are taken as candidates. Every bee should select one node from all the nodes in each stage to be considered as the tag of that word.

For instance, consider the sentence in Figure 1, extracted from the Brown corpus. In this model the value of each node is represented by an integer, which is the index of the related tag as shown in Figure 1.

Word	Tag Index					
	1	2	3	4	5	6
This	QL	DT				
the	AT					
therapist	NN					
may	NNP	MD				
pursue	VB	VBP				
in	RP	NNP	RB	NN	FW	IN
later	RP	RB	JJ	JJR	B	
questioning	VB	JJ	NN			

Fig. 1. Different applicable tag for the words in a sentence "This the therapist may pursue in later questioning" [3]. Underlined tags are the correct ones, according to the Brown corpus.

Constructive moves in forward pass: In every stage, each artificial bee will visit just one node, create partial solution, and collect the nectar according to the weight of its selected node, then returns into the hive. Bees decide to choose a node according to their fragrance. We assume in this paper that the weight of nodes is mutually unequal and proportional to their frequencies. This means that all nodes are not equally interesting for bees to select. If two nodes have equal frequencies, bees would select one in a random manner. Hence, we have assigned a partial function, which is a real-valued weight to each node. The weight denotes the Fragrance value of stage i at node j for the bee, which is computed follows:

$$\text{Fragrance}_i^{\,j} = \frac{\sum freq(s_i, n_j)}{\sum freq(s_i)} \qquad (1)$$

where, $freq(s_i, n_j)$ denotes the number of times the i^{th} word(stage i) occurs with its j^{th} tag (node j). According to the idea of TBL algorithm [9], it is more efficient to choose the most probable tag, which has also the highest frequency for that word in the lexicon. In this paper during each stage, the node selection step is modified. In our method the j^{th} node in the i^{th} stage is chosen randomly by (2).

$$P_i^j = \Pr(\text{node } j \text{ selected as a tag for the word in } i^{th}\text{stage}) = \frac{\text{Fragrance}_i^{\,j}}{\sum_{m=1}^{k_i} \text{Fragrance}_i^{\,m}} \qquad (2)$$

where P_i^j is the probability of selecting node j at stage i by bees and k_i is the maximum number of nodes in stage i. According to equation (1) the most probable tag has the better chance to be selected. However, the worst node in each stage has at least a non-zero probability to be selected as new node. This stochastic nature can aid to avoid local optima. Clearly,

with this scheme good solutions will have better chance than the bad ones. In other words, the preference of an introduced solution depends on the weight of that node. As the weight of the node in each stage increases, the probability of the preferred solution increases proportionally. The stage corresponding to an unknown word is assigned a randomly chosen tag. But a node which appears more often with the given context in the training text has the better chance to be selected. Let us also assume that each constructive move from one node to the next node in each forward pass has the certain weight for the bees while flying along it. The quality of each move is proportional to the partial function, called pollen of that move. The quality of gathered pollen along each move is computed as follow:

$$\text{Pollen}\,(j,j+1)=\frac{freq\,(j,j+1)}{\sum\limits_{j'\in T} freq\,(j,j')} \tag{3}$$

Where, freq (j,j+1) is the number of occurrences of the list of nodes j,j+1 in the training table and T is the set of all possible nodes in this list.

Fragrance and Pollen are considered as the two strong indicators for validating the tags. Flexible merging of them is sufficient for assessing nectar quantity, and can be incorporated as a feature in automatic tagging. Therefore, we combine them into a single weight score via harmonic-mean named convenience of node. In the proposed weighted measure, Fragrance and Pollen of nodes are valued unequally. We use a parameterized weighted harmonic mean of Fragrance and pollen of each node to represent the convenience of node i at stage j. We have assumed that the nectar quantity gathered by the k^{th} bee by choosing node j, located in stage i is computed as follows:

$$\text{convenience}\,(s_{ij})=\frac{\alpha_{Frag}+\alpha_{Poll}}{\frac{\alpha_{Frag}}{\text{Fragrance}\,(s_{ij})}+\frac{\alpha_{Poll}}{\text{Pollen}_{k}{}^{j}}} \tag{4}$$

where, α_{Frag} , is the weight of Fragrance measure, and α_{Poll} is the weight of Pollen measure.

Our experiments have demonstrated that the accuracy of tagging algorithms always receives noticeably more value when the assigned weight to Pollen is higher than Fragrance. Therefore, we treat these parameters with unequal weights when computing the convenience. In this algorithm α_{Frag} and α_{Poll} are set to four and one, respectively, based on previous experiments. In other words, the higher the convenience of the node, the higher the nectar quantity collected along that node. This means that the greatest possible nectar quantity could be collected when flying along the node that has the highest convenience value. In this model, each constructive move in the forward pass consists of choosing a tag of each word in the sentence.

Bee's partial solutions comparison mechanism: All bees return to the hive after generating the partial solutions. In the hive, the bee will participate in a decision making process and compare all generated partial solutions. In this step, every bee makes the decision about abandoning the created partial solution or expanding it in the next forward pass. We assumed that every bee can obtain the information about partial solution quality created by every other bees (the value of the convenience of that node). In order to compare bee's partial solutions, we introduce the concept of sum of the nectar quantity of the partial solution generated by the b-th bee at the stage u in iteration z, which is denoted by $\text{Nect}_b(u,z)$.

$$Nect_{b}(u,z)=\sum_{m=1}^{u}\log(\text{convenience}\,(s_m))\quad b=1,2,...,B \tag{5}$$

Depending on the quantity of the gathered nectar, every bee possesses certain level of loyalty to the tag sequence previously discovered.

Bees use approximate reasoning, and compare their discovered partial solutions with the best as well as the worst discovered partial solution from the start of the search process. In this way, "historical facts" discovered by all members of the bee colony have significant influence on the future search directions. The probability that, b-th bee (at the beginning of a new forward pass at stage u+1 in iteration z) is loyal to the previously discovered partial solution is calculated in the following way:

$$p_{b}\,(u+1,z)=e^{-\frac{Norm-Nect_{\max}(u,z)-Norm-Nect_{b}(u,z)}{uz}},\quad b=1,2,...,B \tag{6}$$

where u is the ordinary number of the forward pass, u=1,2,3,…,U. and $Norm-Nect_b$ represents normalized value of the quantity nectar value of the partial solution discovered by the b^{th} bee.

$$Norm-Nect_{b}(u,z)=\frac{Nect_{\max}(u,z)-Nect_{b}(u,z)}{Nect_{\max}(u,z)-Nect(u,z)_{\min}}\qquad b=1,2,...,B \tag{7}$$

Where $Nect_b$ shows the nectar quantity values of the partial solution discovered by the b-th bee, $Nect_{\max}$ and $Nect_{\min}$ are the nectar quantity values of the best and the worst discovered partial solution from the beginning of the search process, respectively.

Using relation (7) and a random number generator, every artificial bee decides to become an uncommitted follower, or to continue flight along already known path.

The better generated partial solution the higher the probability that the bee will be loyal to the previously discovered partial solution. The greater the ordinary number of the forward pass, the higher the influence of the already discovered partial solution. This is expressed by the term u in the nominator of the exponent. We can see from relation (7) that if a bee has discovered the best tag sequence with the

highest partial fitness value in stage u in iteration z, it will fly along the same partial solution with the probability equal to one. The smaller the fitness value that the bee has discovered, the smaller the probability that the bee will fly again along the same tag. In other words, at the start of the search process bees are "more brave" to search the solution space. The more forward passes they make, the less courage to explore the solution space. The more we are approaching the end of the search process, the more focused the bees are on the already known solutions.

Recruiting process: If at the beginning of a new stage a bee does not want to expand previously creating partial solution, it will go to the dancing area and will follow another bee. Bee dancing area represents the interaction between individual bees in the colony. Within the dance area the bee-dancers (recruiters) "advertise" different partial solutions.

We have assumed that the probability of selecting the advertiser bee's partial solution by any of the uncommitted bees who decide to choose the new partial solution is equal to:

$$P_b = \frac{Nect_b}{\sum_{k=1}^{RC} Nect_k}, \quad b = 1,2,..., RC \tag{8}$$

where $Nect_k$ is the nectar quantity value of the k^{th} advertised solution and RC is the number of recruiters. Using relation (8) and a random number generator, every uncommitted follower joins one bee dancer (recruiter). At the end of this path all bees are free to independently search the solution space and generate the next constructive iteration moves.

III. EXPERIMENTS, ANALYSES AND COMPARISON

A. Used corpus

In this section, the employed corpus and tag set are described. For training and evaluation, two data sets, aiming for broad coverage with different sizes for English language are used. Data sets are gathered from one of the most popular corpora named Brown tagged corpus1. It contains one million words of written American-English texts published in the US in 1961. Nearly 40% of the words appearing in the hand-tagged Brown corpus are ambiguous. We select two different samples of this corpus. The first data set (DS1), which has a medium size, consists of 75 texts, including 38 texts from books and 37 texts from periodicals. The second data set (DS2), which has a large size, contains different categories such as natural, social and behavioral sciences. Table 1 shows the data statistics of these data sets. We have spilt the data sets into 90% "training" and 10% "test" sets. We use 10-fold tests in which 10 different training and test sets with the size of %90 and %10 of corpus respectively, are selected to test the taggers. The tagger is allowed to assign exactly one tag to each word. We measured the tagging performance using a standard measure, namely, accuracy. Accuracy measures the ratio of correct tags (i.e., the proportion of correctly assigned tags to the total number of tokens in the processed corpus) as follows:

$$Accuracy = \frac{\#\,corectly\ tagged\ words}{\#\,Overall\ words} \tag{9}$$

Table 1. Data statistics

Data Set	*DS1*	*DS2*
# of documents	75	155
# of sentences	2346	6850
# of tokens	165489	685900

B. Results and comparison

To demonstrate the superiority of the proposed tagger, it is compared with the conventional and contemporary algorithms previously proposed in the literature. In order to perform a better comparison, performance of the proposed taggers and the other metaheuristic based algorithms, including Genetic algorithm-based (GA) [20, 22, 21, 13], Simulated Annealing-based (SA) [21], Harmony Search based taggers [12], and CHC taggers [21] are evaluated over two different data sets in the same features and experimental conditions. The features we used are the tri-gram transition and lexicon probabilities. The results [21] have shown that Genetic base taggers obtain accuracies as good as typical algorithms such as Viterbi [7] used for stochastic tagging. To compare the algorithms fairly, we have employed the following termination rule. In all of the mentioned algorithms, the current optimal solution is always recorded. For the current optimal solution, we record the number of continuous iterations without improving it. Then we calculate the ratio of this number to the total iteration number. If the ratio exceeds the given upper bound ratio, it means that the continuous running of the algorithm will not result in any improvement to the solution and then the search process ends. In addition, the maximum number of iterations is also given to guarantee that the algorithm will stop after a certain number of searched solutions. It is to be emphasized at this point that the results shown in the rest of the paper are the average over 10 runs of the algorithms, in order to avoid accidental results and to make a fair comparison to be used for drawing conclusions. In addition, to lubricate the comparisons, the algorithms are iterated 500 times in each run since the 500 generations are enough to make converge all of the algorithms. The performance comparison between our proposed tagger with other state-of-the-art evolutionary based taggers is demonstrated in Figure 2. It can be inferred from the results of Figure 2 that proposed algorithm outperforms the other metaheuristic based taggers significantly in all data sets. The obtained results were very competitive when compared with the results of SA [21]. As shown in Figure 2, SA provides worse results than any of the evolutionary algorithms. This proves the advantage of the evolutionary approach. The results obtained for SA are very poor, indicating that SA is not able to solve PoS-tagging problem adequately.

Hidden Markov Model (HMM) [6] and Maximum Entropy (MXPOST) [11] taggers are the well-known statistical tagging

[1] http://khnt.hit.uib.no/icame/manuals/brown/INDEX.HTM

methods and Brill's model taggers is a conventional transformation-based learning (TBL) [9] method [22]. We also define a random and the most frequent tag (MFT) baseline on the Brown corpus. The random baseline is calculated by completely randomly picking one of the tags of each word and it also represents the amount of ambiguity in the corpus. The MFT baseline simply selects the most frequent PoS-tag of each word from the used corpus. If the target word does not exist in the training set, the MFT baseline randomly picks one of the possible tags of the unknown word. Figure 3 presents a comparison between HMM, MXPOST, TBL and our algorithm as well as two standard baselines. As shown in Figure 3, we have achieved some improvements in terms of accuracy. In this way, the heuristic nature of BCO is shown to be useful for tagging where traditional algorithms have low accuracy. As there are very few parameters to be tuned in the proposed algorithm, it is very easy to use. In addition, the behavior of it is very flexible, allowing the size of the context including word and PoS N-gram, to be defined as bigrams or trigram or even higher N-gram. Some classical approaches cannot be applied with two more complex contexts because they are designed to search the data sequence which maximizes the observed data according to a Markov model, i.e., a model in which the current state only depends on the previous one. If we consider tags on the right hand of the word being tagged, our model is not a Markov process anymore. This is a strong reason to further researches with evolutionary based approaches [21]. Another key innovation of the proposed algorithm is the ability of tuning free parameters within the metric in order to optimize them for various languages. It has been successfully applied to English without a priori knowledge other than a tagged corpus. The final advantage is that the greedy search used by the other algorithms can be avoided in the proposed taggers. In each generation of the proposed algorithms a number of possible solutions to the tagging problem are tried. There is the possibility of finding a solution with less effort than a greedy search.

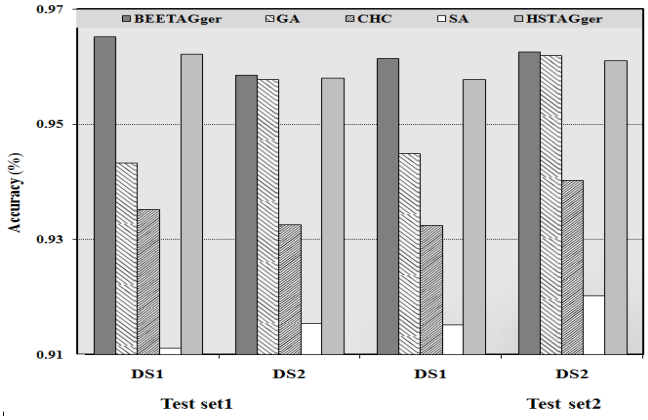

Fig. 2. Comparison of the performance of the metaheuristic taggers.

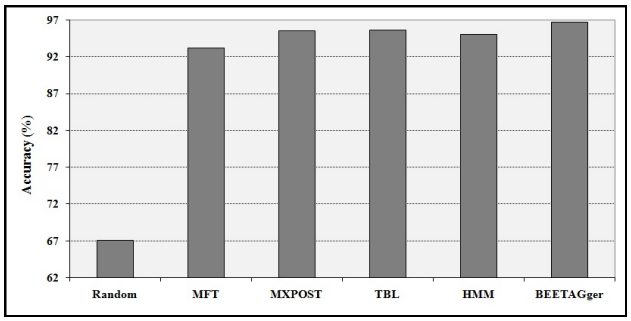

Fig. 3. Results of different methods

IV. CONCLUSION

In this paper the problem of assigning lexical categories to words is studied and a novel algorithm based on Bee Colony Optimization is proposed. The proposed algorithm is designed by modeling tagging problem as an optimization of an objective function. The evaluation of individuals is based on a training table comprised of contexts extracted from an annotated corpus. Our experimental results on different data sets show that the proposed algorithm produces better solutions with high quality considering Accuracy measures in comparison with other known algorithms.

REFERENCES

[1] C. D. Manning, H. Schütze, "Foundations of statistical natural language processing", Cambridge, Mass.: MIT Press, 1999.

[2] S. J. DeRose, "Grammatical category disambiguation by statistical optimization," Comput. Linguist., vol. 14, pp. 31-39, 1988.

[3] W. N. Francis, H. Kucera, "Manual of information to accompany a standard corpus of present-day edited american english, for use with digital computers," Brown University, 1979.

[4] S. R. Tasharofi, F. Oroumchian, M. Rahgozar, "Evaluation of statistical part of speech tagging of persian text," presented at the Signal Processing and Its Applications, 2007.

[5] E. Brill, "Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging," Comput. Linguist., vol. 21, pp. 543-565, 1995.

[6] E. Charniak, "Statistical language learning", Cambridge, Mass.: MIT Press, 1993.

[7] G.D. Forney, "The viterbi algorithm", Proceedings of the IEEE 61 (3) (1973), pp. 268–278.

[8] T. Brants, "TnT: a statistical part-of-speech tagger," presented at the Proceedings of the sixth conference on Applied natural language processing, Seattle, Washington, 2000.

[9] C. Aone, K. Hausman, "Unsupervised learning of a rule-based Spanish Part of Speech tagger," presented at the Proceedings of the 16th conference on Computational linguistics- Volume 1, Copenhagen, Denmark, 1996.

[10] H. V. Halteren, " Improving data driven wordclass tagging by system combination", In the Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistic, vol.1, Montreal, Quebec, Canada, 1998.

[11] Ratnaparkhi.[Online]. Available: ftp://ftp.cis.upenn.edu/pub/adwait/jmx/

[12] R. Forsati, M. Shamsfard, P. Mojtahedpoor "An efficient meta heuristic algorithms for pos-tagging", In the proceedings of The Fifth International Multi-Conference on Computing in the Global Information Technology (ICCGI), Spain, 2010.

[13] K. T. Lua, "Part of speech tagging of chinese sentences using genetic algorithm", presented at the Conference on Chinese Computing, 1996.

[14] D. Teodorovic´, T. DavIdovi´c, T. selmi´c, " Bee colony optimization: the applications survey, ACM Transactions on Computational Logic, pp. 1–20, 2011.

[15] P. Luc˘ic´, D. Teodorovic´, "Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence", In: Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, Sao Miguel, Azores Islands, Portugal, pp. 441–445, 2011.

[16] P. G. Reiser, "Evolution of logic programs: part-of-speech-tagging", Proceedings of the Congress on Evolutionary Computation. pp. 1338-1346, 1999.

[17] G. S. Martin Volk, "Comparing a statistical and a rule-based tagger for German", presented at the In Proceedings of KONVENS-98, Bonn, 1998.

[18] J. R. Curran, R. K. Wong, "Formalisation of transformation based learning", Proceedings of the 2000 Australian Computer Science Conference (ACSC 2000), pp. 51-57, 2000.

[19] A. Ekbal, S. Bandyopadhyay, "Part of speech tagging in bengali using support vector machine", International Conference on Information Technology, 2008.

[20] L. Araujo, "Studying the advantages of a messy evolutionary algorithm for natural language tagging," presented at the Proceedings of the 2003 international conference on Genetic and evolutionary computation: PartII, Chicago, IL, USA, 2003.

[21] E. Alba, et al., "Natural language tagging with genetic algorithms," Information Processing Letters, vol. 100, pp. 173-182, 2006.

[22] L. Araujo, "Symbiosis of evolutionary techniques and statistical natural language processing," Ieee Transactions on Evolutionary Computation, vol. 8, pp. 14-27, 2004.