

ADCSA-WSD: Adapted Discrete Crow Search Algorithm for Word Sense Disambiguation

Bekhouche Abdelaali*, Haouassi Hichem, Rafik Mahdaoui, Rahab Hichem, Ledmi Makhoulouf

ICOSI Lab, Abbes Laghrour University, Khenchela 40004, Algeria

Corresponding Author Email: bakhouche2006@yahoo.fr



<https://doi.org/10.18280/ria.360115>

ABSTRACT

Received: 2 January 2022

Accepted: 10 February 2022

Keywords:

word sense disambiguation, swarm-based optimization, discrete crow search algorithm, natural language processing, combinatorial optimization, swarm-based intelligence

In the field of natural language processing, the semantic disambiguation of words is beneficial to several applications, which helps us to identify the correct meaning of a word or a sequence of words according to the given context. It can be formulated as a combinatorial optimization problem where the goal is to find the set of meanings that contribute to improving the semantic relationship between target words. The Crow Search Algorithm (CSA) is a nature-inspired algorithm. It mimics the food foraging behavior of crow birds and their social interaction. CSA can deal with both continuous and discrete optimization problems. In this paper, the Word Sense Disambiguation (WSD) is modelled as a combinatorial optimization problem that is by nature a discrete problem. For this propose the discrete version of CSA has been adapted for solving the WSD problem and a DCSPA-based WSD approach is proposed and called ADCSPA-WSD. The proposed approach has been evaluated and compared with state-of-the-art approaches using three well-known benchmark datasets (SemCor 3.0, SensEval-02, SensEval-03). Experimental results show that ADCSPA-WSD approach is performing better than other approaches.

1. INTRODUCTION

Semantic disambiguation is an important phase in computational linguistics, which is defined as the ability to identify the sense of ambiguous words according to their context. Word Sense Disambiguation (WSD) typically involves two main tasks: on the one hand, determining the different possible meanings of each word and, on the other hand, tagging each word of a text with its appropriate sense with high accuracy and efficiency.

WSD is usefully applied in many Natural Language Processing (NLP) applications such as sentiment-based text analysis [1], machine translation [2], automatic text summarizer [3], and information extraction [4].

The WSD problem solving methods are divided into two categories; traditional knowledge-based approaches and machine learning based approaches. The machine learning based approaches are further organized into three classes; supervised, unsupervised and semi-supervised methods [5]. The supervised methods based on the largest manually annotated corpus to train while unsupervised knowledge-based methods use dictionaries and lexical resources such as WordNet. The supervised techniques give better results than the unsupervised ones [6]. But the creation of annotated corpora of text needs great effort.

Many researchers have recently focused on the use of meta-heuristic algorithms to determine the best solutions that reflect the best sense [7-9]. These methods rely on a solution branch to explore additional solutions in problem space. This class of algorithms have advanced rapidly in the domain of computational linguistic. The accuracy of population-based algorithms reaches a higher level than single-solution algorithms in WSD.

Several meta-heuristic algorithms inspired from natural

systems have been proposed and widely used in recent years to solve optimization problems with good results compared to traditional techniques. Crow Search Algorithm (CSA) is a newly proposed algorithms, wish was initially introduced by Askarzadeh in 2016 [10]. The main inspiration of CSA algorithm originates from the life of the bird family, called crow. The social intelligence of crow flock and their food gathering process has been the basic motivation for development of this new meta-heuristic optimization algorithm. The main idea of CSA becomes from three principles: crows had a social behavior and live in flock, they memorize the position of their hiding food places, they follow each other to do thievery and protect their caches from being pilfered.

The CSA optimization process is conducted by of flock of n crows started from randomly initial positions, each crow searches the best food position, memorize the position where it hides their exceeding foods and search the hiding food of other crows to thievery it. This process is repeated several iterations and the last position of the best crow represent the optimization problem. CSA is successfully applied in many areas such as benchmark optimization [11-13], medical data preservation [14], feature selection [15, 16], medical image segmentation [17], optimal design of water distribution networks [18], Real structural design problems [12], classification in data mining [19].

Discrete Crow Search Algorithm (DCSPA) is a discrete version of CSA that is successfully applied for solving optimization problems [20]. Therefore, on observing the superior performance of DCSPA, we would like to adapt its application in solving WSD problem.

Recently, the WSD problem is presented as an optimization problem. Ajeena and Chinmayan [7] have proposed a definition to solve WSD as a combinatorial optimization

problem, which has motivated us in this paper to use DCSA to solve this problem. In order to disambiguate words in context, we use the DCSA to find the best sense for ambiguous words using the gloss-context overlap algorithm (Lesk) as an objective function to perform the local measurement. To the best of our knowledge, this is the first work that uses the discrete version of CSA algorithm to solve the WSD problem. The main motivations of choosing the DCSA are their best results obtained is several optimization problems, their simple equations and easy implementation, by which we may obtain promising results when solving WSD optimization problems.

The remainder of this paper is structured as follows: in section 2, we briefly discuss some of the related works in the area of WSD; in section 3, we present the methodology employed to develop our approach; and finally, we present the experimental results of the proposed approach.

2. RELATED WORKS

This section is a literature review of previous works related to this research paper. Previous studies of WSD problem can be approximately divided into two categories: traditional knowledge-based approaches, which exploit the external lexical knowledge to address this issue such as machine-readable dictionaries, thesauri, ontologies, etc.; and supervised WSD approaches uses machine-learning techniques for inducing a classifier from manually sense-annotated data sets.

The first category of study assigns a sense to an ambiguous word by comparing each of its senses with those of the surrounding context [21]. Typically, some semantic similarity metric is used for calculating the relatedness among senses. Several methods have been reported in the literature to address this issue based on traditional knowledge. The most cited work is the approach of Kwon et al. [22], this approach has three steps, firstly it extracted the semantic relationships from the lexical knowledge for creating the word vector representation, secondly it analysed the text for creating the contextual words and ambiguous words vectors and finally it computed the similarity between the words vectors. Another study for WSD problem is presented by Alkhatlan et al. [23], it is based on a word embedding method using the WordNet database. In this work, the authors computed word semantic similarity while giving thought to a multiple stemming algorithm. Pasini and Navigli introduce in reference [24] a Train-O-Matic approach to solve WSD problem using a dictionary-based and language-independent approach in order to annotate millions of instances with word sense.

The drawbacks of these research methods stem from the pairwise comparisons between senses, and thus the number of computations grows exponentially with the number of words. that is, for a sequence of m words where each has up to n meanings, they need to consider up to n^m meaning sequences.

The second category is a supervised WSD which include machine-learning techniques to learn a classifier from labeled training texts of corpora, that is, sets of examples encoded in terms of a number of features together with their appropriate sense label (or class); Generally, supervised approaches to WSD have obtained better results than Traditional knowledge-based methods

Among the most cited is the work of Fahandezhi et al. [25] who proposed a new approach based on supervised learning to identify the correct meaning in standard English lexical sample tasks. The authors have assigned vector coefficients for

a more accurate context representation and to tackle the problem of unbalanced datasets they used the reduction process to find the best feature from independence corpora. Abed et al. [26] proposed an approach based on a Stanford dependencies generator for WSD. The authors used harmony search algorithm to maximize the overall semantic similarity of the set of words that it has been parsed by dependency generator. The experimental result shows that this technique is able to produce effective solutions for most instances of the used datasets.

However, we argue these methods suffers from certain weaknesses: they require manually annotated training data for each term that needs to be disambiguated. but manual annotation is an expensive, difficult and time-consuming process which is not practical to apply on a large scale.

Recently, many meta-heuristic methods have been applied for WSD due to the success attain in the literature. These algorithms are defined as an iterative approach which design to find a heuristic (partial search method) that may give a good solution to an optimization problem. These types of algorithms have rapidly progressed in the domain of computational linguistics.

The accuracy of population-based approaches reaches a grade better than that of single-solution methods in WSD [8]. Various meta-heuristic methods including genetic algorithm, Ant colony algorithm, particle swarm optimization, cuckoo algorithm, optimisation firefly and bees algorithm were applied to word sense disambiguation problem. Among them we can mention the work of Al-Saiagh et al. [8], which developed a hybrid meta-heuristic method. They have used the particle swarm optimization and simulated annealing algorithm to determine the global best sense of a given context. They used extended Lesk algorithm as an objective function. To determine the correct sense in a particular context, Bakhouché et al. have used at the local level the similarity measure between word senses, and at the level global the combinatorial optimization algorithm [27, 28]. The authors adopt ant colony algorithm and compared it to simulated annealing and genetic algorithms. The ant colony algorithm obtained better precision, recall and execution time.

The hybrid method introduced by Alsaeedan et al. [29] that consist of self-adaptive genetic algorithm and ant colony one. The authors [29, 30] used unlabelled corpus to determine the semantic category of the ambiguous word. This technique has also been used by Hausman et al. [31, 32] to find the optimal set of definitions across several sentences from SemCor. Then, the researcher compares the algorithm to other WSD algorithms. Rajini and Vasuki [9] has developed several algorithms (cuckoo algorithm, optimisation firefly and bees algorithm) intended to annotate samples and a large amount of data in the used corpus. They have evaluated these algorithms using the standard SemEval2016 task 11 dataset. The results show that the optimisation firefly approach is performing better than other algorithms. Another study based on a combinatorial version of particle swarm optimization algorithm for solving WSD problem is proposed by Ajeena and Chinmayan [7]. The experimental result shows that the algorithm is very powerful and can perform better than the other algorithms.

These algorithms achieve high accuracy in disambiguation in many domains and many languages corpus. Recently DCSA as a metaheuristic was successfully used in optimization problems and gives good results, which we motivated to apply it for WSD problem.

3. PROPOSED APPROACH

The proposed method is unsupervised method, which uses WordNet for solving the WSD problem. We model the WSD as a combinatorial optimization problem, in which, the proposed approach finds senses of target words at the same time. In the proposed approach, the text to be disambiguated is restructured as a discrete vector, and it consists of the following three main phases:

Phase 1: Text pre-processing includes the following tasks: segmentation, word tokenization, punctuation removal, number removal, case folding, and stop-word removal;

Phase 2: in this phase, the WordNet is used to provide the number of senses for the ambiguous words, the semantic relations and glosses for each ambiguous word of the processed text;

Phase 3: semantic disambiguation using the DCSA algorithm.

The general description of the proposed approach is presented in Figure 1.

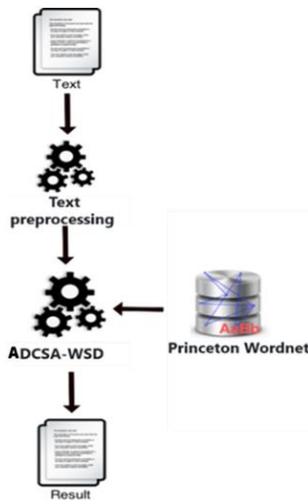


Figure 1. Flowchart of the proposed ADCSA–WSD approach for WSD problem

3.1 Discrete crow search algorithm for the word sense disambiguation problem

3.1.1 Discrete Crow Search Algorithm

The original CSA is a nature-inspired metaheuristic algorithm that simulates the searching behaviour of the crows to the food [10]. Briefly, crows live in flocks and are considered to be among the most intelligent animals, they can memorize faces, communicate in sophisticated ways and intelligently hide and retrieve food, and they take solid safety measures to hide their excess food and stealing other's food [10].

The CSA is proposed for optimizing continuous problems [30] but it cannot be applied directly for discrete optimization problems, for this, DCSA is proposed recently by Ledmi et al. [20] and successfully used for the Mining Quantitative Association Rules problem as a discrete optimization problem.

In DCSA, a flock of crows (particles) cooperate to find optimal solutions to optimization problems. When applying a DCSA algorithm, the problem is represented as a combinatorial optimization problem where the position of a crow represents a possible solution of the problem. At the start of the DCSA, a number N of crows (flock size) are initially

spread out on the search space. After that, as all swarm-based algorithms, a number of iterations are repeated until a termination condition is met (maximum number of iterations or expected quality of the obtained solution). Each crow i has a position at time it (iteration) in the search space specified by a vector and a memory used for storing his hiding place represented respectively in Eqns. (1) and (2).

$$X_i^{i,it} = [x_1^{i,it}, x_2^{i,it}, \dots, x_d^{i,it}] \quad (1)$$

$$M_i^{i,it} = [m_1^{i,it}, m_2^{i,it}, \dots, m_d^{i,it}] \quad (2)$$

where, ($i = 1, 2, \dots, N$; $it = 1, 2, \dots, iter$; d represent the problem dimensionality). An objective value is assigned for a position of each crow, which is obtained via an objective function to be improved. The best position obtained by all the flock at the end of the search is considered as a best solution to the optimization problem.

In each iteration, the crows move to new positions for searching an eventual better nourishment source using the following strategy: Each crow i at each iteration it chooses randomly a crow j from their flock to followed their best hiding place (M_j^{it}). The current position of a crow i is updated according to the Eqns. (3) and (4).

$$X_i^{it+1} = \begin{cases} \varpi(X_i^{it}, M_i^{it}, int(S_i^{it})) & r_i \geq AP_i^{it} \\ a \text{ random position} & \text{otherwise} \end{cases} \quad (3)$$

$$S_i^{it} = fl_i^{it} \times nonZHD(M_j^{it}, X_i^{it}) \quad (4)$$

where, X_i^{t+1} indicates the new position of the crow i at iteration $it+1$, M_i^{it} denotes the memorized position of the crow j at iteration it and r_i is random vector in $[0,1]$, AP_i^{it} is the awareness probability of the crow i at iteration it , fl_i^{it} is the flight length of the crow i at iteration it , $\varpi()$ is a permutation operator calculated using the Algorithm 1, $int()$ returns the integer value of a real value and $nonZHD()$ is a Non- Zero Hamming Distance calculates the Hamming distance between two vectors.

Algorithm 1. Permutation Algorithm

Input: Vectors $X[]$, $M[]$ and S

Output: $Y[]$

begin

$Y=X$

for ($I=1$; $i \leq S$; $i++$)

$j = \mathbf{rand}(1; \text{Size}(X))$

$Y[j] = M[j]$

End for

end

3.1.2 ADCSA for the WSD problem

In this section we explain how the DCSA is adapted and applied for solving the WSD problem.

a) Position encoding and initialisation

Each crow in the flock explores the search space and generates a disambiguation solution to a text composed of a set of words. A disambiguation solution is an assignment of a sense to each word in the input text; each word w_i has a synset Syn_i of size T_i in WordNet $Syn_i = \{S_i^1, S_i^2, \dots, S_i^{T_i}\}$ where S_i^j is a discrete value represents the code of the j^{th} sense of the i^{th}

word in the disambiguated text.

ADCSA is used to find an optimal set of senses, which maximizes overall similarity, for this, each solution is represented by a crow's position. Then, a position at iteration it denoted by a vector $X^{i,it} = [x_1^{i,it}, x_2^{i,it}, \dots, x_d^{i,it}]$, where $x_j^{i,it}$ takes a discrete value from Syn_j of the j^{th} word in the disambiguated text, d is the number of words to be disambiguated together.

For example, let a text composed of the following four words: w_1, w_2, w_3 and w_4 to be disambiguated, the word w_1 has three senses $Syn_1 = \{1, 2, 3\}$; w_2 has two senses $Syn_2 = \{1, 2\}$

w_3 has four senses $Syn_3 = \{1, 2, 3, 4\}$ and w_4 has three senses $Syn_4 = \{1, 2, 3\}$.

A sample of position (solution) vector is represented in Figure 2. This solution means that the second sense is assigned for w_1 , the first sense is assigned for w_2 , the fourth sense is assigned for w_3 and the second sense is assigned for w_4 .

W ₁	W ₂	W ₃	W ₄
2	1	4	2

Figure 2. A sample of a crow's position encoding

Algorithm 2. Pseudo code of ADCSA for WSD

Input: T: Text to be disambiguated, P: the population size, AP: the awareness probability, fl: the flight length, t_{\max} is the number of iterations

Output: Disambiguation vector of the words in the text T.

```

Initialize the crow's positions X randomly
Evaluate the fitness of all crow positions using Equation (6)
Initialize the memory position M of all crows as these initial positions
for (t=1 to  $t_{\max}$ )
  for (i=1 to Size) do
    Choose a random index j
    Generate a random number R in [0,1]
    if  $R \geq AP_j^t$  then
      Calculate  $S_i^t$  as in Equation (4)
       $X_i^{t+1} = \otimes (X_i^t, M_j^t, S_i^t)$ ,
      Evaluate the  $\otimes$  operator as in the Algorithm (1)
    else
       $X_i^{t+1}$  = a random position from the search space
    end if
    Check the feasibility of the new solution  $X_i^{t+1}$ 
    Evaluate the new position of a crow  $Fit(X_i^{t+1})$  using Equation (6)
    if ( $Fit(X_i^{t+1}) \geq Fit(M_i^t)$ )
      Updates the crow's memory  $M_i^{t+1} = X_i^{t+1}$ 
    end if
  end for
end for
Obtain the best memory M from all crows.
Consider M as the disambiguation vector of words in the input text T

```

end

b) Fitness function

In the swarm-based metaheuristics, the quality of each solution (position) must be evaluated using a fitness function. In this work, for a crow position of length n , the fitness value is calculated using the function defined in Eq. (6). The used fitness function attempted to maximise the overall relatedness for the given text.

The similarity between words is defined by formula (5):

If S_1 Senses (w_1) and S_2 Senses (w_2):

$$ScoreLesk(S_1, S_2) = |gloss(S_1) \cap gloss(S_2)| \quad (5)$$

where, $gloss(S_i)$ is the set of words built from the textual dictionary definition of word w_i . The meaning of an ambiguous word depends on the best result of the Lesk score. However, this requires computing the gloss overlaps between Senses (w_1) and Senses (w_2). A new modified version of this algorithm uses the highly interconnected set of relations amongst the synonym's hyponymy and hypernymy that WordNet offers. Let us denote $Syn(S)$ as the set of synonyms of sense S through an explicit link in WordNet; then, the extended-level measure between sense S_1 and sense S_2 ,

denoted $Ext_{Lesk}(S_1; S_2)$, is the following:

$$Ext_{Lesk}(S_1; S_2) = \frac{|D(S_1) \cup (Syn(S_1)) \cap D(S_2) \cup (Syn(S_2))|}{|D(S_1) \cup (Syn(S_1)) \cap D(S_2) \cup (Syn(S_2))|} \quad (6)$$

4. EXPERIMENTAL RESULTS

The performance of the proposed ADCSA-WSD approach was evaluated on three well-known corpus in the WSD research field, and compared with four state-of-the-art WSD approaches, including GA [27], GA-LS [32], HAS [29] and H-PSO [8]. The proposed approach was implemented in Python. The performance comparison was based on three metrics consisting of precision, recall and F-measure. This section starts with a description of the three used measures. It then discusses the lack of a general standard in word sense disambiguation evaluations and how SemCor and SenseEval attempt to address this issue. Finally, it reports the result obtained by the proposed approach on SemCor and SenseEval followed by an analysis of the results and comparison them with state-of-the-art approaches.

4.1 Evaluation criteria

The results described in this paper are expressed in terms of precision, recall, and F-measure. This section briefly explains these evaluation metrics in the context of WSD, precision and recall can be given by Eqns. (7) and (8) respectively.

$$\text{Precision}(P) = \frac{\text{Number of correctly disambiguated words}}{\text{Total number of words disambiguated}} \quad (7)$$

$$\text{Recall}(R) = \frac{\text{Number of correctly disambiguated words}}{\text{Number of words that should have been disambiguated}} \quad (8)$$

Both precision and recall assume values in the interval [0,1].

$$F - \text{measure} = 2 \times \frac{\text{Number of correctly disambiguated words}}{\text{Total number of words disambiguated} + \text{Number of words that should have been disambiguated}} \quad (9)$$

4.2 WSD dataset

The proposed ADCSA-WSD approach was tested on three well-known corpora consists of SemCor 3.0, SensEval-2 and SensEval-3. In Table 1, the main characteristics of each one is presented.

SemCor 3.0 [33]: is an English corpus consists the texts semantically annotated. It was automatically generated by mapping of WordNet 1.6 to WordNet 3.0 senses. It can be one of the largest publicly available Sense-Tagged resources. All the words in this corpus are tagged by part of speech and by sense. SemCor was manually annotated.

SensEval-02 [34]: the Second International conference on Evaluating Word Sense Disambiguation Systems was held on July 5-6, 2001. A main goal of SENSIVAL-2 was to encourage new languages to participate. It evaluated WSD systems on some types of tasks (All-words, lexical sample, Translation) on 12 languages (English, Czech, Estonian, Dutch, Swedish, Basque, Spanish, Italian, Korean, Japanese).

SensEval-03 [35]: the SensEval-3 evaluation exercise took place in the third international conference on the evaluation of semantic text analysis systems on March 1, 2004 - April 15, 2004. It included fourteen different tasks for word sense disambiguation, multilingual annotations, identification of semantic ideas, subcategorization acquisition and logic forms.

4.3 Parameters setting of the proposed ADCSA-WSD

In this sub-section, the effect of the two main sensitive parameters' values (population size and number of iterations) of ADCSA-WSD performance is studied, so we test different values of these parameters.

In the first scenario, the population size parameter is tested on several values in the range between 10 and 500 and the number of iterations is fixed. For each value of the population size, in Figure 3 the ADCSA-WSD performance in terms of F-measure is reported. From Figure 3 as can be seen, with an increase in the number of particles of the algorithm, the quality of solutions in terms of F-Measure is improved. However, it is clear that if the number of particles passes 250 the F-Measure value is fixed in the maximum value 70. So, the best value for the population size is selected as 250 particles for the rest of experiments.

In the second scenario, the population size parameter is fixed and the number of iterations is tested on several values in the range between 1 and 100. For each value of iteration

These metrics can be misleading if both precision and recall are not considered together. For example, a method with a very high precision of 1 but a low recall of 0.1 means that the system covered only 10% of all words although all the disambiguated words are correct. In order to have a balance between precision and recall, a third metric called F-measure is used. This measure is calculated as the harmonic mean of the two measures recall and precision, giving each the same importance. It allows a system to be evaluated taking both the precision and recall into account using a single score, which is helpful when describing the performance of the system and in comparing methods.

F-measure can be expressed by the formula in Eq. (9).

number, the ADCSA-WSD performance in terms of F-measure is reported in Figure 4, in which the variation of the F-measure value vs. number of iterations is presented. From Figure 4 as can be seen, with an increase in number of iterations of the algorithm, the quality of solutions in terms of F-Measure is improved. But it is clear that if the number of iterations passes 50 the F-Measure value is fixed in the maximum value 70. So, the best values of the iteration number are selected for the rest of experiments as 50.

4.4 Comparison results

To present the efficiency of the proposed method, we have reported a comparison with the four approaches GA, GA-LS, HAS and H-PSO on three well-known corpora in the field of WSD problem SemCor 3.0, SensEval-2 and SensEval-3. The compared approaches are selected because are recent approaches for the WSD problem and were used similar research methods as the proposed approach, i.e., meta-heuristic algorithms. In Table 1 the characteristics of the three used corpora are presented.

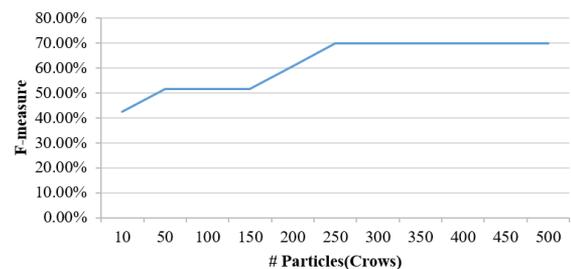


Figure 3. Variation of the F-Measure value vs. number of particles (Crows)

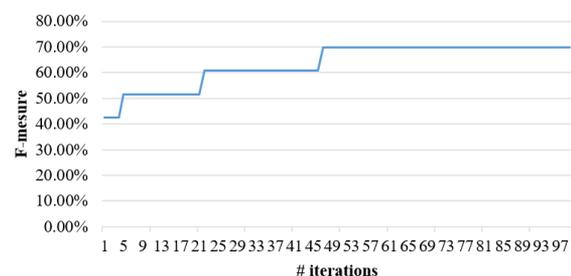


Figure 4. Variation of the F-Measure value vs. number of iterations

Table 1. Lexical statistics of the three used corpuses

Corpus	# Sentences	# Lemmas	# Instances	Inventory	Text sources
SemCor 3.0*	352	23,346	234,113		
SensEval-02	242	2282	17251	WordNet	80% Brown corpus, 20% a novel, The Red Badge of Courage
SensEval-03	352	1850	17773		

Resource: *<https://web.eecs.umich.edu/~mihalcea/senseval/data.html>

The proposed ADCSA-WSD needs to the four following parameters (Population size, number of iterations, Flight length and Awareness probability). The last two parameters values are inspired from the original work of DCSA[20] and are fixed to 0.2 and 0.5 respectively, however the population size and the number of iterations are generally sensitive to the fields of application. Then, their two parameters are studied in the sub-section 4.3 and the best values are selected. The ADCSA-WSD parameters used in the rest of experiments are presented in Table 2.

Table 2. Parameter values used in the ADCSA-WSD algorithm

Parameters	Definitions	Values
Population size	Population size	250
AP	Awareness probability	0.5
Fl	Flight length	0.2
Dimension	Dimension of vector	40
Number of iterations	Max number of iterations	50

Figures 5, 6 and 7 show the performance evaluation of the proposed approach and existing algorithms on each of the corpus used in this study based on the three measures presented above.

In Figure 5 a comparison of the proposed approach and other approaches on the SemCor corpus is presented. The best value of F-Measure is 71.79% that is obtained by the proposed approach whereas the F-Measure values of the compared approaches varies from 60.15% obtained by GA and 65.35% obtained by GA-LS. The other metrics are also higher than those obtained by the compared approaches, the best precision on SemCor is 73.68% obtained by the proposed approach, whereas the precision values of the compared approaches vary from 62.38% obtained by GA and 67.44% obtained by H-POS.

Figure 6 shows the results obtained on SensEval-2 corpus. From Figure 6, it is clear that, the proposed approach performs better than other approaches. The percentage of F-Measure obtained by our approach is 68.29, while TSP-ACO, HAS, SA-GA, and H-PSO approaches attain respectively 62.90%, 60.69%, 51.49% and 65.83%. For the precision measure, our approach reported 70% while the best of the compared approaches reported 66.04%. Similarly, to above measures, the best value of recall attained is 66.67% obtained by our approach while the best recall obtained by the other approaches is 65.62%.

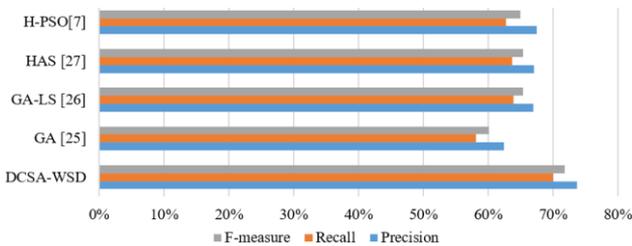


Figure 5. Comparison of results of ADCSA-WSD and related works based on SemCor 3.0 corpus

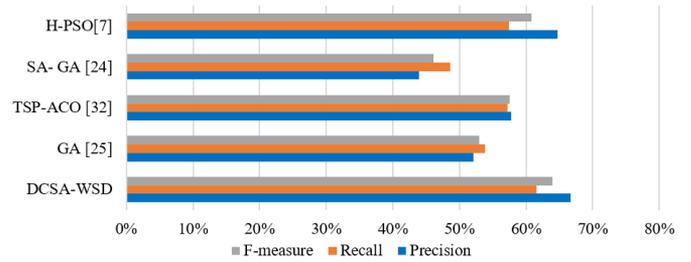


Figure 6. Comparison of results of ADCSA-WSD and related works based on SensEval-2 corpus

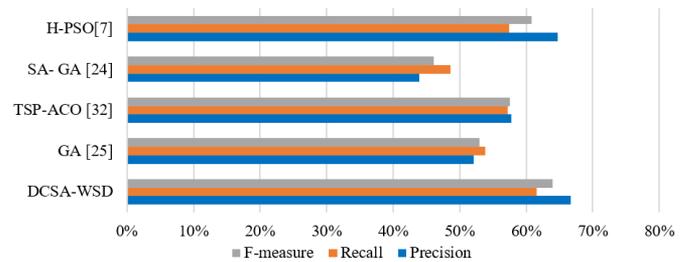


Figure 7. Comparison of results of ADCSA-WSD and related works based on SensEval-3 corpus

The comparison on SensEval-3 is showed in Figure 7. The proposed approach in this study attained better results compared to the other approaches both in terms of F-measure, precision and recall. The proposed approach attained F-Measure value of 68.29% while GA, TSP-ACO, SA-GA and H-PSO obtained respectively 52.95%, 57.50%, 46.15 and 60.84%. The proposed approach attained 66.67% and 61.54% as a precision and recall values respectively that is better by almost 3% and 4% than the best values obtained by the H-PSO approach.

The result indicates that the proposed ADCSA-WSD approach is reliable and can disambiguate ambiguous words effectively.

The Figures 5-7 represent the comparison results of the proposed approach and four state-of-art approaches on the three used corpora. These figures show that the proposed approach outperformed all the compared algorithms on all used corpora in terms of precision, recall and F-Measure. From these figures, it is clear that the precision decreases by 6.24%, 3.91%, and 1.99% by using the proposed approach in relation to the best compared approach on SemCor, SensEval-2 and SensEval-3 respectively. For the recall measure the proposed approach obtains better result, and the recall value is increased by 6.15%, 1.05% and 3.16% on SemCor, SensEval-2 and SensEval-3 respectively compared to the best results obtained by the compared approaches. The F-Measure result is increased by 6.42%, 2.46% and 3.16% on SemCor, SensEval-2 and SensEval-3 respectively.

The performance of this approach based mainly on the semantic measure, which does not generally give best accuracy. Therefore, the use of the gloss-context overlaps approach (LESK) to compute the disambiguation of words in

the local context in one hand and the propagation of the local measures at the global level by using ADCSA metaheuristic in another hand provides good balancing between the semantic measure property, and the accuracy.

However, other approaches need linguistic resources or knowledge tagged data during the measurement process to evaluate the semantic measure, which takes a long time to evaluate the sense of words. In this study, we focus on the semantic optimization problem rather than knowledge-based disambiguation.

Hence, we compared our approach to the similar approaches that used standard semantic similarity and relatedness measures.

5. CONCLUSION

WSD is the process of determining which sense of word ambiguous in a given context. In this article, a knowledge-based unsupervised approach for solving WSD issue is proposed. We modelled the WSD problem as a combinatorial optimization problem, in which the set of ambiguous words are disambiguated together. This research used the new metaheuristic algorithm (ADCSA) as an optimization method to identify a combination of senses for ambiguous words. The proposed approach utilizes the gloss-context overlap algorithm (Lesk) as an objective function to perform the local measurement. The proposed approach is evaluated on three well-known corpuses SemCor 3.0, SensEval-2 and SensEval-3 and the experiments we carried out show that the combination of Lesk algorithm as objective to be optimized and ADCSA as optimization algorithm, produced good disambiguation results.

Based on the obtained results by ADCSA-WSD, future research needs to be carried out to improve the performance of proposed approach. One such attempt could be to try modifications of ADCSA and to explore the possibility of using other meta-heuristic algorithms in WSD problem.

REFERENCES

[1] Mukherjee, S. (2021). Sentiment analysis. IML.NET Revealed, pp. 113-127. https://doi.org/10.1007/978-1-4842-6543-7_7

[2] Poibeau, T. (2017). Machine Translation. MIT Press.

[3] Zaman, F., Shardlow, M., Hassan, S.U., Aljohani, N.R., Nawaz, R. (2020). HTSS: A novel hybrid text summarisation and simplification architecture. *Information Processing & Management*, 57(6): 102351. <https://doi.org/10.1016/j.ipm.2020.102351>

[4] Grishman, R. (2015). Information extraction. *IEEE Intelligent Systems*, 30(5): 8-15. <http://doi.org/10.1109/MIS.2015.68>

[5] Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2): 10. <https://doi.org/10.1145/1459352.1459355>

[6] Navigli, R., Litkowski, K.C., Hargraves, O. (2007). Semeval-2007 task 07: Coarse-grained English all-words task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic, pp. 30-35.

[7] Ajeena Beegom, A.S., Chinmayan, P. (2020). Solving word sense disambiguation problem using combinatorial

PSO. *Journal of Intelligent & Fuzzy Systems*, 38(5): 6193-6200. <http://doi.org/10.3233/JIFS-179701>

[8] Al-Saiagh, W., Tiun, S., Al-Saffar, A., Awang, S., Al-Khaleefa, A.S. (2018). Word sense disambiguation using hybrid swarm intelligence approach. *PloS One*, 13(12): e0208695. <https://doi.org/10.1371/journal.pone.0208695>

[9] Rajini, S., Vasuki, A. (2021). Word sense disambiguation using optimisation techniques. *International Journal of Cloud Computing*, 10(1-2): 78-89. <http://doi.org/10.1504/IJCC.2021.113986>

[10] Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures*, 169: 1-12. <https://doi.org/10.1016/j.compstruc.2016.03.001>

[11] Abdalh, G.Y., Algamal, Z.Y. (2020). A QSAR classification model of skin sensitization potential based on improving binary crow search algorithm. *Electronic Journal of Applied Statistical Analysis*, 13(1): 86-95. <https://doi.org/10.1285/i20705948v13n1p86>

[12] Gholami, J., Mardukhi, F., Zawbaa, H.M. (2021). An improved crow search algorithm for solving numerical optimization functions. *Soft Computing*, 25(14): 9441-9454. <http://doi.org/10.1007/s00500-021-05827-w>

[13] De Souza, R.C.T., dos Santos Coelho, L., De Macedo, C.A., Pierzan, J. (2018). A V-shaped binary crow search algorithm for feature selection. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, Rio de Janeiro, Brazil, pp. 1-8. [IEEE.http://Doi.org/10.1109/CEC.2018.8477975](http://doi.org/10.1109/CEC.2018.8477975)

[14] Mandala, J., Rao, M.C.S. (2019). Privacy preservation of data using crow search with adaptive awareness probability. *Journal of Information Security and Applications*, 44: 157-169. <https://doi.org/10.1016/j.jisa.2018.12.005>

[15] Manikandan, P., Selvarajan, S. (2013). A hybrid optimization algorithm based on cuckoo search and PSO for data clustering. *International Review on Computers and Software*, 8(9): 2278-2287.

[16] Sayed, G.I., Hassanien, A.E., Azar, A.T. (2019). Feature selection via a novel chaotic crow search algorithm. *Neural Computing and Applications*, 31(1): 171-188. <https://doi.org/10.1007/s00521-017-2988-6>

[17] Meraihi, Y., Gabis, A.B., Ramdane-Cherif, A., Acheli, D. (2021). A comprehensive survey of crow search algorithm and its applications. *Artificial Intelligence Review*, 54(4): 2669-2716. <https://doi.org/10.1007/s10462-020-09911-9>

[18] Fallah, H., Kisi, O., Kim, S., Rezaie-Balf, M. (2019). A new optimization approach for the least-cost design of water distribution networks: Improved crow search algorithm. *Water Resources Management*, 33(10): 3595-3613. <http://doi.org/10.1007/s11269-019-02322-8>

[19] Hichem, H., Rafik, M., Ouahiba, C. (2022). New discrete crow search algorithm for class association rule mining. *International Journal of Swarm Intelligence Research (IJSIR)*, 13(1): 1-21. <http://doi.org/10.4018/IJSIR.2022010109>

[20] Ledmi, M., Moumen, H., Siam, A., Haouassi, H., Azizi, N. (2021). A discrete crow search algorithm for mining quantitative association rules. *International Journal of Swarm Intelligence Research (IJSIR)*, 12(4): 101-124. <http://doi.org/10.4018/IJSIR.2021100106>

[21] Agirre, E., López de Lacalle, O., Soroa, A. (2014).

- Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1): 57-84. http://dx.doi.org/10.1162/COLI_a_00164
- [22] Kwon, S., Oh, D., Ko, Y. (2021). Word sense disambiguation based on context selection using knowledge-based word similarity. *Information Processing & Management*, 58(4): 102551. <http://dx.doi.org/10.1016/j.ipm.2021.102551>
- [23] Alkhatlan, A., Kalita, J., Alhaddad, A. (2018). Word sense disambiguation for Arabic exploiting Arabic wordnet and word embedding. *Procedia Computer Science*, 142: 50-60. <https://doi.org/10.1016/j.procs.2018.10.460>
- [24] Pasini, T., Navigli, R. (2020). Train-o-Matic: Supervised word sense disambiguation with no (manual) effort. *Artificial Intelligence*, 279: 103215. <https://doi.org/10.1016/j.artint.2019.103215>
- [25] Fahandezi Sadi, M., Ansari, E., Afsharchi, M. (2019). Supervised word sense disambiguation using new features based on word embeddings. *Journal of Intelligent & Fuzzy Systems*, 37(1): 1467-1476. <http://doi.org/10.3233/JIFS-182868>
- [26] Abed, S.A., Tiun, S., Omar, N. (2015). Harmony search algorithm for word sense disambiguation. *PloS One*, 10(9): e0136614. <https://doi.org/10.6084/m9.figshare.1509913>
- [27] Bakhouch, A., Yamina, T., Schwab, D., Tchechmedjiev, A. (2015). Ant colony algorithm for Arabic word sense disambiguation through English lexical information. *International Journal of Metadata, Semantics and Ontologies*, 10(3): 202-211. <https://doi.org/10.1504/IJMSO.2015.073880>
- [28] Schwab, D., Goulian, J., Tchechmedjiev, A., Blanchon, H. (2012). Ant colony algorithm for the unsupervised word sense disambiguation of texts: Comparison and evaluation. In *Proceedings of COLING 2012*, Mumbai, India, pp. 2389-2404.
- [29] Alsaeedan, W., Menai, M.E.B., Al-Ahmadi, S. (2017). A hybrid genetic-ant colony optimization algorithm for the word sense disambiguation problem. *Information Sciences*, 417: 20-38. <https://doi.org/10.1016/j.ins.2017.07.002>
- [30] Alsaeedan, W., Menai, M.E.B. (2015). A self-adaptive genetic algorithm for the word sense disambiguation problem. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Seoul, South Korea, pp. 581-590. https://doi.org/10.1007/978-3-319-19066-2_56
- [31] Hausman, M. (2011). A genetic algorithm using semantic relations for word sense disambiguation (Doctoral dissertation, University of Colorado at Colorado Springs).
- [32] Abed, S.A., Tiun, S., Omar, N. (2016). Word sense disambiguation in evolutionary manner. *Connection Science*, 28(3): 226-241. <https://doi.org/10.1080/09540091.2016.1141874>
- [33] Petrolito, T., Bond, F. (2014). A survey of wordnet annotated corpora. In *Proceedings of the Seventh Global WordNet Conference*, pp. 236-245.
- [34] Edmonds, P., Cotton, S. (2001). SensEval-2: Overview. In *Proceedings of SensEval -2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pp. 1-5.
- [35] Rada, M., Edmonds, P. (2004). *Proceedings of SensEval -3: The third international workshop on the evaluation of systems for the semantic analysis of text*. Barcelona, Spain.