Word Sense Disambiguation Using an Evolutionary Approach

Mohamed El Bachir Menai
Department of Computer Science, College of Computer and Information Sciences
King Saud University, P.O.Box 51178, Riyadh 11543, Saudi Arabia
menai@ksu.edu.sa
http://faculty.ksu.edu.sa/menai

Keywords: evolutionary algorithms, genetic algorithms, natural language understanding, word sense disambiguation

Received: July 22, 2013

Word sense disambiguation is a combinatorial problem consisting in the computational assignment of a meaning to a word according to a particular context in which it occurs. Many natural language processing applications, such as machine translation, information retrieval, and information extraction, require this task which occurs at the semantic level. Evolutionary computation approaches can be effective to solve this problem since they have been successfully used for many NP-hard optimization problems. In this paper, we investigate main existing methods for the word sense disambiguation problem, propose a genetic algorithm to solve it, and apply it to Modern Standard Arabic. We evaluated its performance on a large corpus and compared it against those of some rival algorithms. The genetic algorithm exhibited more precise prediction results.

Povzetek: Razločitev pomena besed je v tem prispevku izpeljana z evolucijskim pristopom.

1 Introduction

Ambiguity is a key feature of natural languages. That is, words can have different meanings (polysemy), depending on the context in which they occur. Humans deal with language ambiguities by acquiring and enriching common sense knowledge during their lives. However, solving computationally the ambiguity of words is a challenging task, since it relies on knowledge, its representation, extraction, and analysis. In Arabic language, ambiguity is present at many levels [30], such as homograph, internal word structure, syntactic, semantic, constituent boundary, and anaphoric ambiguity. The average number of ambiguities for a token in Modern Standard Arabic (MSA) is 19.2, whereas it is 2.3 in most languages [30].

Word sense disambiguation (WSD) is a challenging task in the area of natural language processing (NLP). It refers to the task that automatically assigns the appropriate sense, selected from a set of pre-defined senses for a polysemous word, according to a particular context. Indeed, the identification of one word sense is related to the identification of neighboring word senses. WSD is necessary for many NLP applications and is believed to be helpful in improving their performance such as machine translation, information retrieval, information extraction, part of speech tagging, and text categorization. WSD has been described as an AI-complete (Artificial Intelligence-complete) problem [53] that is analogous to NP-complete problems in complexity theory. It can be formulated as a search problem and solved approximately by exploring the solution search space using heuristic and meta-heuristic algorithms. Several approaches have been investigated for WSD in occidental languages (English, French, German, etc.), including knowledge-based approaches and machine learning-based approaches. However, research on WSD in Arabic language is relatively limited [5,6,17,24–27,38].

Evolutionary algorithms (EAs) are search and optimization methods inspired by biological evolution: natural selection and survival of the fittest in the biological world. Several types of EAs were developed, including genetic algorithms (GAs) [41], evolutionary programming (EP) [32], evolution strategies (ES) [69,76] and genetic programming (GP) [45]. EAs are among the most popular and robust optimization methods used to solve hard optimization and machine learning problems. They have been widely and successfully applied in several real world applications [55] and research domains. These include NLP research, such as query translation [20], inference of context free grammars [43], tagging [8], parsing [7], and WSD [22,35,84]. Araujo [9] has written a survey paper on how EAs are applied to statistical NLP, which is highly recommended.

In this paper, we study the potential of GAs in formulating and solving the WSD problem, apply them to MSA, and compare them with some existing methods. We implemented and experimented different variants of GAWSD (GA for Arabic WSD) resulting in the introduction of a competitive approach for WSD. The rest of the paper is organized as follows. The next section presents a brief overview of EAs. Section 3 contains a brief introduction to WSD, and presents the main approaches to solve it. Section 4 describes Arabic language peculiarities and challenges. Section 5 presents the proposed approach to WSD, and describes in detail the proposed algorithm. Section 6 reports the test results, and Section 7 discusses them. Finally, Sec-

tion 8 concludes this paper and emphasizes some future directions.

2 Evolutionary algorithms

EAs are built around four key concepts [21]: population(s) of individuals competing for limited resources, dynamic changing populations, suitability of an individual to reproduce and survive, and variational inheritance through variation operators.

EAs are categorized as "generate and test" algorithms that involve growth or development in a population of chromosomes in genotype space (individuals containing genes) of candidate solutions in phenotype space (real features of an individual). An evaluation function called the *fitness function*, defined from chromosome representation, measures how effective the candidate solutions are as a solution to the problem. Variation operators such as *recombination* (or *crossover* in case of recombination of two parents) and *mutation* are applied to modify the individual content and promote diversity.

Algorithm 1: Evolutionary Algorithm

```
Initialize P(1);
t ← 1;
while not exit criterion do
    evaluate P(t);
    selection;
    recombination;
    mutation;
    survive;
    t ← t + 1;
```

The basic steps of an EA are outlined in Algorithm 1. An *initial population*, $P(1)$, is randomly generated and a selection process (*selection*) is then performed to select parents based on the fitness of individuals (*evaluate*). The *recombination* and *mutation* operators are applied on parents to obtain a population of offspring. The population is renewed (*survive*) by selecting individuals from the current population and offspring for next generation $(t+1)$. This evolutionary process continues until a termination condition, *exit criterion*, is reached.

GAs [41] are the most traditional EAs which are based on biological genetics, natural selection, and emergent adaptive behavior. They are associated to the use of binary, integer, or real valued vectors for the chromosome representation. The crossover and mutation are the genetic operators. The crossover is the main operator (applied with a high probability), and the mutation is the secondary one (applied with a low probability). The main steps of a GA are outlined in Algorithm 2 [15]. GP [45] can be considered as an extension of GAs in which each individual is a computer program represented by a rooted tree. In this case, the fitness function determines how well a program is able to solve the problem.

Algorithm 2: Genetic algorithm

input : $Population_{size}$, $Problem_{size}$, $P_{crossover}$, $P_{mutation}$
output: S_{best}

```
Population ← Initialize(Population_size, Problem_size);
Evaluate(Population);
S_best ← BestSolution(Population);
while not exit criterion do
    Parents ← SelectParents (Population);
    Children ← φ;
    for Parent_1, Parent_2 ∈ Parents do
        (Child_1, Child_2) ← Crossover (Parent_1, Parent_2, P_crossover);
        Children ← Mutate (Child_1, P_mutation);
        Children ← Mutate (Child_2, P_mutation);
    Evaluate(Children);
    S_best ← BestSolution(Children);
    Population ← SelectToSurvive (Population, Children);
Return(S_best)
```

3 Classification methods for word sense disambiguation

WSD can be described as the task of assigning the appropriate sense to all or some of the words in the text. More formally, given a text T as a sequence of words or bag of words $\{w_1, w_2, \cdots, w_k\}$, the WSD problem asks to identify a mapping A from words w_i to senses $Senses_D(w_i)$ encoded in a dictionary D. $A(w(i))$ is the subset of the senses of w_i which are appropriate in the context T [62].

A WSD system includes mainly four elements: word senses selection, external use of knowledge resources, context representation, and selection of automatic classification method. The first element, selection of word senses, is concerned with the sense distinction (sense inventory) of a given word. The second element, external knowledge sources, involves a repository of data consisting of words with their senses. Two main kinds of resources are distinguished: structured resources and unstructured resources. The third element of WSD is concerned with the representation of the context that aims to convert unstructured input text into a structured format to become suitable for automatic methods. The last element of WSD is the choice of the classification method. The key distinction between classification methods depends on the amount of knowledge and supervision quantified into them.

In the following, we survey the main classification methods used for WSD, as they represent a key issue in designing a WSD system.

Classification methods can be achieved using different

approaches [62]: knowledge-based and machine learning-based approaches. Knowledge-based methods rely on external lexical resources, such as dictionaries and thesauri, whereas machine learning methods (supervised, unsupervised, or semi-supervised methods) rely on annotated or unannotated corpus evidence and statistical models. Other methods use both corpus evidence and semantic relations. They can be further categorized as token-based or type-based approaches. While token-based approaches associate a specific meaning with each occurrence of a word depending on the context in which it appears, type-based disambiguation is based on the assumption that a word is consensually referred with the same sense within a single text [62].

Other approaches have been considered, such as word sense dominance-based methods [46,54,59], domain-driven disambiguation [37], and WSD from cross-lingual evidence [33].

3.1 Knowledge-based methods for word sense disambiguation

Several knowledge-based methods for WSD have been proposed, including gloss-based methods, selectional preferences-based methods, and structural methods.

Gloss based-methods consist in calculating the overlap of sense definitions of two or more target words using a dictionary. Such methods include the well-known Lesk algorithm [50] and one of its variants proposed by Banerjee and Pedersen [11].

Selectional preferences (or restriction) based methods exploit association provided by word-to-word, word-to-class, or class-to-class relations to restrict the meaning of a word occurring in a context, through grammatical relations [62]. Several techniques have been proposed to model selectional preferences, such as selectional associations [70,72], tree cut models [51], hidden Markov models [1], class-based probability [2,19], and Bayesian networks [18]. An application of such associations to expanding an Arabic query of a search engine [5] shows that the performance of the system can be increased by adding more specific synonyms to the polysemous terms.

Structural approaches are semantic similarity-based methods and graph-based methods. The main idea behind these approaches is to exploit the structure of semantic networks in computational lexicons like WordNet [31], by using different measures of semantic similarity. Some examples of knowledge-based systems include Degree [61] and Personalized PageRank [3].

Similarity-based methods are applicable to a local context, whereas graph-based methods are applicable to a global context. Similarity-based methods select a target word sense in a given context based on various measures of semantic similarity, such as those introduced by Rada et al. [68], Sussna [77], Leacock and Chodorow [47], Resnik [71], Jiang and Conrath [42], and Lin [52]. Elghamry [27] proposed coordination-based semantic similarity for disambiguating polysemous and homograph nouns in Arabic, based on the assumption that nouns coordinating with an ambiguous noun provide bootstraps for disambiguation.

Graph-based methods select the most appropriate sense for words in a global context using lexical chains (sequence of semantically related words by lexicosemantic relations) [62]. Many computational models of lexical chains have been proposed, including those of Hirst and St-Onge [40], Galley and McKeown [34], Harabagiu et al. [39], Mihalcea et al. [57], and Navigli and Velardi [63].

3.2 Machine leaning methods for word sense disambiguation

There are three classes of machine learning methods: supervised, unsupervised, and semi-supervised methods. All of them have been largely applied to WSD.

The most popular supervised WSD methods include decision lists [82], decision trees [60], naïve Bayes classifiers [66], artificial neural networks [60,79], support vector machines [29,48,85], and ensemble methods [28]. Farag and Nürnberger [6] used a naïve Bayes classifier to find the correct sense for Arabic-English query translation terms by using bilingual corpus and statistical co-occurrence.

Unsupervised WSD methods usually select the sense from the text by clustering word occurrences. Through measuring the similar neighboring words, new occurrences can be classified into clusters/senses. Since unsupervised methods do not use any structured resource, their assessment is usually difficult. The main approaches to unsupervised WSD are context clustering [67,75], word clustering [14,65], and co-ocurrence graphs [80]. Diab [25] introduced an unsupervised method called SALAAM (stands for Sense Annotations Leveraging Alignments And Multilinguality) to annotate Arabic words with their senses from an English WordNet using parallel Arabic-English corpus based on translational correspondences between Arabic and English words. Lefever et al. [49] used a multilingual WSD system, called ParaSense, where the word senses are derived automatically from word alignments on a parallel corpus.

To address the lack of the training data problem, semi-supervised WSD methods use both annotated and unannotated data to build a classifier. The main semi-supervised WSD methods are based on a bootstrapping process which starts with a small amount of annotated data (called seed data) for each word, a large corpus of unannotated data, and one or more classifiers. The seed data are used to train the classifier using a supervised method. This classifier then uses the unannotated data to increase the amount of annotated data and decrease the amount of unannotated data. This process is repeated until achieving an amount threshold of unannotated data. Co-training and self-training are two bootstrapping approaches used in WSD. Co-training uses two classifiers for local and global information (e.g. [56]). Self-training uses only one classifier that merges the two types of information. An example of self-training ap-

proach is illustrated by Yarowsky algorithm [83].

3.3 Evolutionary algorithms for word sense disambiguation

Gelbukh et al. [35] used a GA as a global optimization method (the total word relatedness is optimized globally) to tackle WSD problem. An individual is represented by a sequence of natural numbers of possible word senses retrieved from a dictionary, and the Lesk measure [50] is used to evaluate its fitness. The experimental results obtained on Spanish words show that this method gives better results than existing methods which optimize each word independently.

Decadt et al. [22] used a GA to improve the performance of GAMBL, a WSD system. WSD is formulated as classification task distributed over word experts. A memory-based learning method is used to assign the appropriate sense to an ambiguous word, given its context. The feature selection and algorithm parameter optimization are performed jointly using a GA. The experimental results obtained on Senseval-3 English all-words task, show the constructive contribution of the GA on system performance with a mean accuracy of 65.2%.

Zhang et al. [84] proposed a genetic word sense disambiguation (GWSD) algorithm to maximize the semantic similarity of a set of words. An individual is represented by a sequence of natural numbers of possible word senses retrieved from WordNet. The length of the chromosome is the number of words that need to be disambiguated. The fitness function used is based on the Wu-Palmer similarity measure [81] in which the domain information and the frequency of a given word sense are included. The evaluation of the algorithm gives a mean recall of 71.96%.

4 Arabic language

4.1 Arabic language characteristics

Arabic language belongs to the Afro-Asian language group. Its writing is right to left, cursive, and does not include capitalization. Arabic letters change shape according to their position in the word, and can be elongated by using a special dash between two letters.

The language is highly inflectional. An Arabic word may be composed of a stem plus affixes (to refer to tense, gender, and/or number) and clitics (that include some prepositions, conjunctions, determiners, and pronouns). Words are obtained by adding affixes to stems which are in turn obtained by adding affixes to roots.

Diacritization or vocalization in Arabic, consists in adding a symbol (a diacritic) above or below letters to indicate the proper pronunciation and meaning of a word. The absence of the diacritization in most of Arabic electronic and printed media poses a real challenge for Arabic language understanding. Arabic is a pro-drop language: it allows subject pronouns to drop, like in Italian, Spanish, Chinese, and Japanese [30].

Dealing with ambiguity in Arabic is considered as the most challenging task in Arabic NLP. There are two main levels of Arabic ambiguity [10,30]: (1) Homographs are words that have the same spelling, but different meanings. The main cause of homographs is due to the fact that the majority of digital documents do not include diacritics; (2) Polysemy is the association of one word with more than one meaning. Ambiguity in Arabic can be also present in other levels, such as: internal word structure ambiguity, syntactic ambiguity, semantic ambiguity, constituent boundary ambiguity, and anaphoric ambiguity [30].

MSA is the subject of this research. It is the language of modern writing and formal speaking. It is the language universally understood by Arabic speakers around the world. In contrast, Classical Arabic (CA) is the language of religious teaching, poetry, and scholarly literature. MSA is a direct descendent of CA [12].

4.2 Arabic text preprocessing

Text preprocessing consists in converting a raw text file into a well-defined sequence of linguistically-meaningful units, such as characters, words, and sentences [64]. It includes the following tasks:

- Tokenization or sentence segmentation is the process of splitting the text into words.
- Stop-word removal is the process of filtering a text from the stop-words, such as prepositions and punctuation marks, assuming that they do not deeply alter the meaning of the text.
- Stemming is the process of removing prefixes and suffixes to extract stems.
- Rooting is the process of reducing words to their roots.

There are some well-known algorithms for morphological analysis, such as Khoja's stemmer [44], Buckwalter's morphological analyzer [16], the Tri-literal root extraction algorithm [4], MADA (Morphological Analysis and Disambiguation for Arabic) [38,73], and AMIRA [23].

5 Proposed approach

Amongst the various methods presented in Section 3, the mostly used methods for WSD are supervised WSD and knowledge-based methods. Supervised WSD methods achieve better performance than knowledge-based methods given large training corpora, but they are generally limited to small contexts. Knowledge-based methods can exploit all available knowledge resources, such as dictionaries and thesauri, but they require exponential computational time as the number of words increases. Our approach consists in approximating solutions to WSD problem by using GAs

to improve the performance of a gloss-based method. We adopt a similar individual (or chromosome) representation to the one presented in [8,35], but different evaluation functions of the individual fitness and different selection methods. To the best of our knowledge, there is no published research proposing an evolutionary computing-based approach to solve the WSD problem in Arabic language.

Algorithm 3 outlines the main steps of the genetic algorithm for Arabic WSD (GAWSD).

A text T is transformed into a bag of words $\{w_1, w_2, \cdots, w_k\}$ in a preprocessing phase, including stop-word removal, tokenization, and rooting. The accuracy of Arabic WSD algorithms can be increased by reducing the words to their root form. A morphological analysis is then needed to extract the root form of the word. The comparative evaluation of Arabic language morphological analyzers and stemmers [74], namely Khoja's stemmer, the tri-literal root extraction algorithm, and the Buckwalter morphological analyzer, shows that Khoja's stemmer achieves the highest accuracy. In our algorithm, Khoja's stemmer is used to reduce words to their roots. The senses $Senses_{AWN}(w_i)$ of each word w_i are retrieved from Arabic WordNet (AWN) [13] as word definitions which are reduced in turn to bags of words. An GA is used to find the most appropriate mapping from words w_i to senses $Senses_{AWN}(w_i)$ in the context T. The best individual S_{best} returned by the GA, is decoded into the phenotype space to obtain the appropriate sense of words $WordsSense_{best}$.

Algorithm 3: GAWSD

input : T, k, $Population_{size}$, $P_{crossover}$, $P_{mutation}$
output: $WordsSense_{best}$

$\{w_1, w_2, \cdots, w_k\} \leftarrow$ Preprocessing(T);
for $i = 1, k$ **do**
  $Definitions(w_i) \leftarrow$ AWN(w_i);
  $Senses_{AWN}(w_i) \leftarrow$ Preprocessing($Definitions(w_i)$);
$S_{best} \leftarrow$ GA($Population_{size}$, k, $Senses_{AWN}(w_i)_{i=1,k}$, $P_{crossover}$, $P_{mutation}$);
$WordsSense_{best} \leftarrow$ Decode(S_{best});
Return($WordsSense_{best}$)

To formulate the WSD problem in terms of GA, we need to define the following elements:

- A representation of an individual of the population.
- A method to generate an initial population.
- An evaluation function to determine the fitness of an individual.
- A description of the genetic operators (crossover and mutation).
- Methods to select parents for the mating pool and individuals to survive to the next generation.
- Values for the several algorithm parameters (population size, crossover and mutation rates, termination condition, tournament size, etc.).

More specifically, we propose the following formulation to solve the WSD problem. Alternative solutions for key elements of the algorithm, such as generation of initial population, fitness function, etc., will be considered to find out the appropriate resolution.

- An individual Ind_p represents a possible sequence of sense indexes assigned to the words in the context T. It is represented by a fixed-length integer string $Ind_p = \{SI^l(w_1), SI^m(w_2), \cdots, SI^r(w_k)\}$, where each gene $SI^j(w_i)$ is an index to one of possible senses of the word w_i: $SI^0(w_i), SI^1(w_i) \cdots SI^l(w_i) \cdots$.
- The initial population is generated according to one of the following schemes:
  - Random generation: The value of each gene of an individual is selected randomly from 1 to $SenseNum$ using the uniform distribution, where $SenseNum$ is the number of possible senses for the corresponding word.
  - Constructive generation: All the senses of a given word are distributed in a round-robin way to the corresponding gene of individuals in the population.
- The fitness function is measured by the word sense relatedness. Two different measures are considered: the Lesk measure [50] and one of its variants, called the extended Lesk measure. The Lesk measure calculates the sense which leads to the highest overlap between the sense definitions of two or more words. Formally, given two words w_1 and w_2, and their respective senses $Senses_{AWN}(w_1)$ and $Senses_{AWN}(w_2)$, for each two senses $S_1 \in Senses_{AWN}(w_1)$ and $S_2 \in Senses_{AWN}(w_2)$, the Lesk measure is defined by Equation 1,

$$score_{Lesk}(S_1, S_2) = |gloss(S_1) \cap gloss(S_2)| \quad (1)$$

where $gloss(S_i)$ represents the bag of words corresponding to the definitions of the sense S_i.
The extended Lesk measure calculates the overlap between the sense definitions of a target word and the words in its context. Formally, given a word w and its $context(w)$, for each sense S_i of w, the extended Lesk measure is defined as by Equation 2,

$$score_{extendedLesk}(S_i) = |context(w) \cap gloss(S_i)| \quad (2)$$

where $gloss(S_i)$ represents the bag of words corresponding to the definitions of the sense S_i.

- A single-point crossover operator combines two individuals (parents) to generate two new ones (children or offspring). The crossover point is chosen randomly.

- A single-point mutation operator creates a new individual by randomly changing the value of a selected gene in an individual. The new value of the gene is selected randomly from 1 to $SenseNum$, where $SenseNum$ is the maximum number of possible senses for the corresponding word.

- Two parent selection methods are considered: the *roulette wheel* (or *fitness proportionate*) and the *tournament selection* methods. The Sigma scaling function can be used with the roulette wheel selection method to make the GA less susceptible to premature convergence. It is described as follows:

$$ExpVal(i,t) = \begin{cases} 1 + \frac{Fitness(i) - \overline{Fitness}(t)}{2 \cdot \sigma(t)} \\ \text{if } \sigma(t) \neq 0 \\ \\ 1.0 \\ \text{otherwise} \end{cases} \tag{3}$$

$$RWS(i,t) = \frac{ExpVal(i,t)}{\sum_{j=1}^{n} ExpVal(j,t)} \tag{4}$$

where $ExpVal(i,t)$ is the expected value of individual i at iteration t, $RWS(i,t)$ is the probability of individual i to be selected by the roulette wheel at iteration t, $\overline{Fitness}(t)$ is the mean fitness of the population at iteration t, $\sigma(t)$ is the standard deviation of the population fitnesses at iteration t, and n is the population size.

- The *elitist* survivor selection method is considered as a combination between a generational and steady state schemes. The best sequence is then retained unaltered at each generation, which is found generally to significantly improve the performance of GAs.

- Two termination conditions are considered: number of generations and number of fitness evaluations.

6 Experiments

In this section, we present results of experiments with GAWSD on an Arabic data corpus used in [38,78]. It contains 1132 text documents collected from Arabic newspapers [1] from August 1998 to September 2004. This corpus was used for Arabic classification tasks. It contains 6 different categories of documents (arts: 233 documents, economics: 233 documents, politics: 280 documents, sports: 231 documents, woman: 121 documents, and information technology: 102 documents). In our experiments, we selected 60 documents (10 documents from each class) from which we collected 5218 words. With support of linguists, the corpus was manually sense-tagged using AWN. The corpus contains about 48528 sense-tagged instances, which gives an average number of senses per word of 9.3. Two groups of annotators were asked to select the sense for the target word they find the most appropriate in each sentence. The selection of a sense for a target word was made from a list of senses given by AWN. The agreement rate for a target word was estimated as the number of sentences which are assigned identical sense to the target word by the two groups of annotators over the total number of sentences containing the target word. The average inter-annotator agreement gave a score of 91%.

We considered words within a text window to limit the context size (e.g. a window size of 2 means that the context of every word contains at most 5 words, including the target word). These data were used to evaluate the performance of GAWSD under different settings and to compare it with a naïve Bayes classifier. All the results were averaged over 100 runs, and the sense proposed by the algorithm was compared to the manually selected sense.

6.1 Performance evaluation criteria

The performance evaluation criteria were based on the number of True positives (TP), True negatives (TN), False positives (FP), and False negatives (FN).

The fitness evaluation criteria were as follows. The best fitness value $maxFitness$ and its occurrence number nb($maxFitness$) were recorded in each run. The mean fitness $\overline{Fitness}$ and its standard deviation $\sigma(Fitness)$ were calculated over 100 runs.

The performance evaluation criteria were as follows.

1. The precision P is the percentage of correct disambiguated senses for the ambiguous word: $P = TP/(TP+FP), TP+FP \neq 0$.
2. The recall R is the number of correct disambiguated senses over the total number of senses to be given: $R = TP/(TP+FN), TP+FN \neq 0$.
3. The fall-out F is the number of incorrect disambiguated senses over the total number of incorrect senses: $F = FP/(FP+TN), FP+TN \neq 0$.

The results are shown as convergence graphs of the algorithms in respective experiments (parameter setting, impact of parent selection methods, impact of fitness evaluation function, performance evaluation, and comparison of the algorithms). Detailed results of performance comparison are also reported in Tables in terms of mean precision $\overline{P}$, mean recall $\overline{R}$, and mean fall-out $\overline{F}$, along with their respective standard deviations $\sigma(P)$, $\sigma(R)$, and $\sigma(F)$ over the corpus.

The next Section presents the results of experiments conducted on GAWSD for parameters' tuning.

[1] ElAhram: http://www.ahram.org.eg/,
ElAkhbar: http://www.akhbarelyom.org.eg/,
and ElGomhoria: http://www.algomhuria.net.eg/

6.2 Selection of the parameters

The choice of the parameters is critical for the performance of GAs. The first set of experiments involves the numerical investigation of the GA parameters and their impact on the performance of GAWSD, namely, Population size $Population_{size}$, Crossover rate $P_{crossover}$, Mutation rate $P_{mutation}$, and Termination condition $T_{condition}$.

In all experiments on parameters tuning, we used the same following settings: a window size of 2, a random initialization of the population, the roulette wheel as a parent selection method, the Lesk measure as a fitness evaluation function, and $T_{condition} = 50$ generations (except when varying the termination condition).

6.2.1 Variation of population size

We studied the effect of population size on the performance of GAWSD. We chose $P_{crossover} = 0.9$, $P_{mutation} = 0.1$, and made the population size varying from 6 to 100. As shown in Figure 1 (a), the number of best fitness is increasing with the size of the population, but its value is relatively constant. $Population_{size} = 50$ is a good compromise between the number of best fitness and mean fitness.

6.2.2 Variation of crossover and mutation rates

The performance of GAs is always sensitive to the genetic operators' rate. In order to study the effect of varying $P_{crossover}$ and $P_{mutation}$ on the performance of GAWSD. We carried out experiments on the test suite while setting $Population_{size} = 50$. The results are presented in Figure 1 (b,c). The average best results were obtained with $P_{crossover} = 0.70$ and $P_{mutation} = 0.15$.

6.2.3 Variation of termination condition

We studied the performance of GAWSD according to the number of fitness evaluations which we made varying from 1000 to 10,000. The other parameters $Population_{size}$, $P_{crossover}$, $P_{mutation}$ were fixed to 50, 0,70, and 0.15, respectively. Figure 1 (d) shows the results obtained. As expected, the number of best fitness is increasing with the number of fitness evaluations. However, the best fitness value is, in average, more or less indifferent to the number of fitness evaluations starting from 4000.

6.3 Effect of the initial population generation and sigma scaling function

The results of this section are intended to show the combined effect of the method used to generate the initial population (denoted Rnd: randon generation; Cve: constructive generation) and the sigma scaling function (denoted Sig) as a smoothing function for the roulette wheel selection method.

The best fitness is given by all the four variants: random or constructive generation of the initial population, with or without control of the selection pressure of parents (sigma scaling). The highest number of best fitness and best mean fitness are given when the initial population is generated randomly without using the sigma scaling function (Rnd). The second best mean fitness is also obtained with the random generation of the population, when the sigma scaling function is applied ($RndSig$).

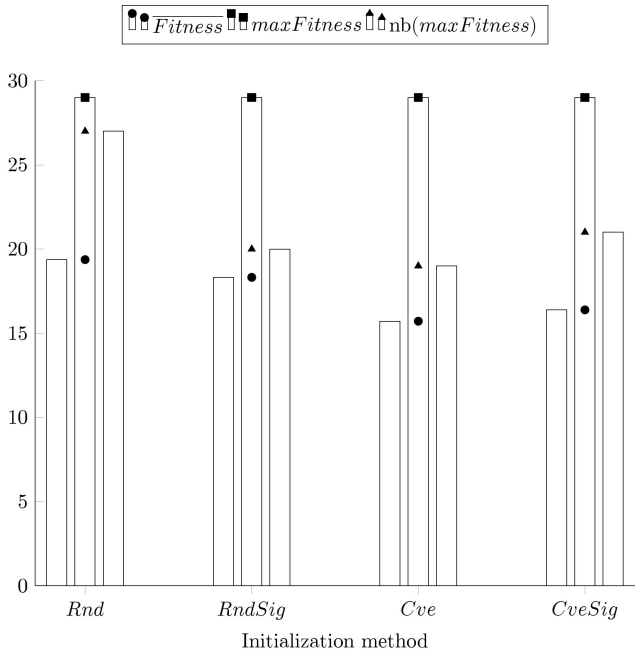


Figure 2: Effect of the initial population generation and sigma scaling function. The abbreviations Rnd, $RndSig$, Cve, and $CveSig$ stand for random generation, random generation + sigma scaling, constructive generation, and constructive generation + sigma scaling, respectively. The best, mean, and number of best fitness values are depicted over 100 runs.

A conclusion can be drawn about the parameter settings. The overall results presented in Figures 1 and 2, substantiate our choice of the following parameters and initialization method for the next experiments: $P_{crossover} = 0.70$, $P_{mutation} = 0.15$, $Population_{size} = 50$, and $T_{condition} \geq 4000$ fitness evaluations. The random generation of the initial population without sigma scaling function is chosen, since it gave substantial improvement with respect to the other schemes.

6.4 Effect of parent selection method

We first investigated the sensitivity of the tournament selection method to variations of tournament size by experimenting with different tournament sizes ($k = 10\% \ldots 40\%$ of the population size $Population_{size}$). Results, presented in Figure 3 (a), show how the best fitness and its mean change with the tournament size. However, the number of best fitness is, in average, constant. The overall best results were obtained with $k = 20$.

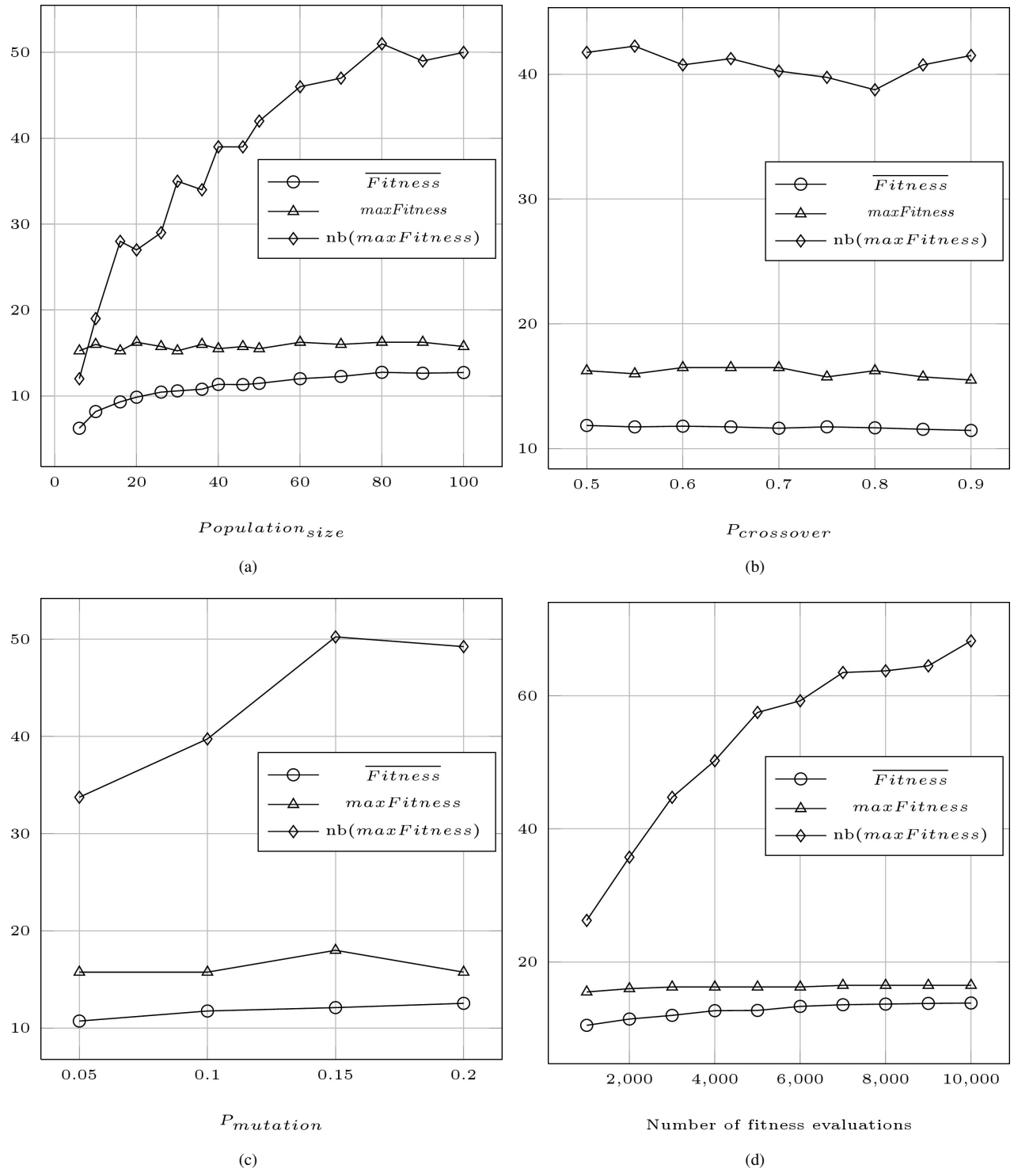
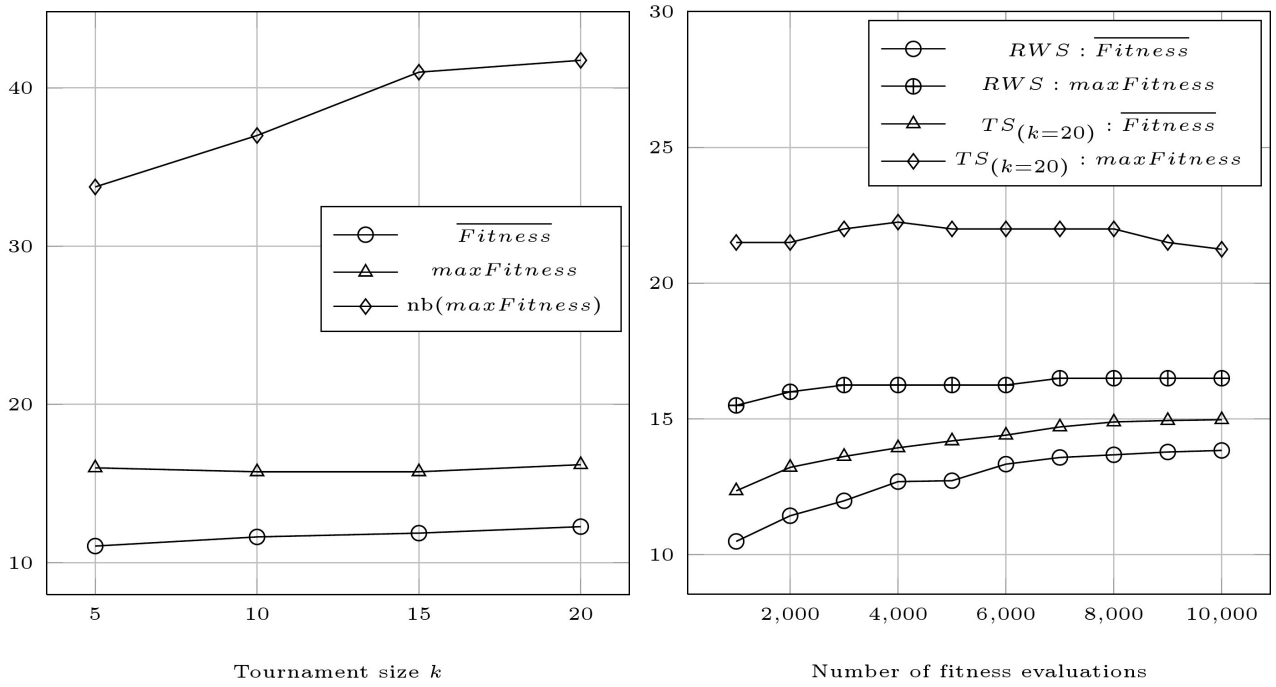


Figure 1: Selection of the parameters for the algorithm GAWSD. The best, mean, and number of best fitness values are depicted over 100 runs. (a) Variation of the population size $Population_{size}$. (b) Variation of the crossover rate $P_{crossover}$. (c) Variation of the mutation rate $P_{mutation}$. (d) Variation of the termination condition $T_{condition}$.



(a) Variation of the tournament size k.

(b) Parent selection methods.

Figure 3: Comparison of the roulette wheel selection (RWS) and tournament selection (TS). The best, mean, and number of best fitness values are depicted over 100 runs.

We then compared the two parent selection methods: roulette wheel selection (RWS) and tournament selection (TS). The tournament size k was fixed at 20. Figure 3 (b) shows that for the same number of fitness evaluations, the best fitness and its mean obtained with $TS_{k=20}$, were always better than those achieved with RWS.

6.5 Sensitivity to fitness evaluation function

The results of the experiments presented in this section are intended to show the influence of the fitness evaluation function on the performance of GAWSD. We compared four variants of GAWSD based on the parent selection method (RWS or $TS_{k=20}$) and relatedness measure ($Lesk$ or $extendedLesk$): $RWS\&Lesk$, $RWS\&extendedLesk$, $TS_{k=20}\&Lesk$, and $TS_{k=20}\&extendedLesk$.

The graphs of Figure 4 show the results obtained in terms of mean precision $\overline{P}$. The overall best results were obtained with the Lesk measure and the tournament selection method.

6.6 Performance evaluation

To investigate the sensitivity of GAWSD to variations of target word context, we experimented with different text window sizes ($W_{size} = 1 \ldots 5$). The Lesk measure and tournament selection were adopted in GAWSD (GAWSD$_{TS}$). The superiority of the tournament selection method on the roulette wheel selection is shown in Section 6.4.

Two baseline algorithms, Random and FirstSense, were implemented to compare the performance of GAWSD$_{TS}$. Random algorithm selects randomly a sense for each word among its senses given by AWN. FirstSense algorithm selects the first sense that appears in the list of senses for each word.

Results, presented in Figure 5, show how the performance of GAWSD$_{TS}$ changes with the text window size for a given maximum number of fitness evaluations (set to 4000) and how it compares to the baseline algorithms. Convergence graphs of Figure 5 show that GAWSD$_{TS}$ performs better than the baseline algorithms in terms of mean recall and mean precision. The performance improvement of the algorithm is more substantial with the increase in text window size. This behavior was expected, since adding new words to the context of a target word, results in reducing the set of potential word senses, and hence the size of the search space.

6.7 Comparison with other methods

Existing methods related to Arabic WSD, as presented in Section 3, include SALAAM [25], naïve Bayes classifiers [6], and coordination-based semantic similarity [27]. However, their results cannot be compared directly to our algorithms' results, since those methods are related to different WSD tasks and their results were generated on dif-

Table 1: Performance comparison of the algorithms GAWSD$_{TS}$, naïve Bayes (NB), Random, and FirstSense.

Algorithm	$\overline{P}$	$\sigma(P)$	$\overline{R}$	$\sigma(R)$	$\overline{F}$	$\sigma(F)$
GAWSD$_{TS}$	**0.79**	**0.08**	0.63	0.29	**0.20**	**0.12**
NB	0.66	0.21	**0.68**	**0.24**	0.32	0.31
Random	0.38	0.35	0.31	0.42	0.59	0.40
FirstSense	0.54	0.22	0.48	0.30	0.42	0.37

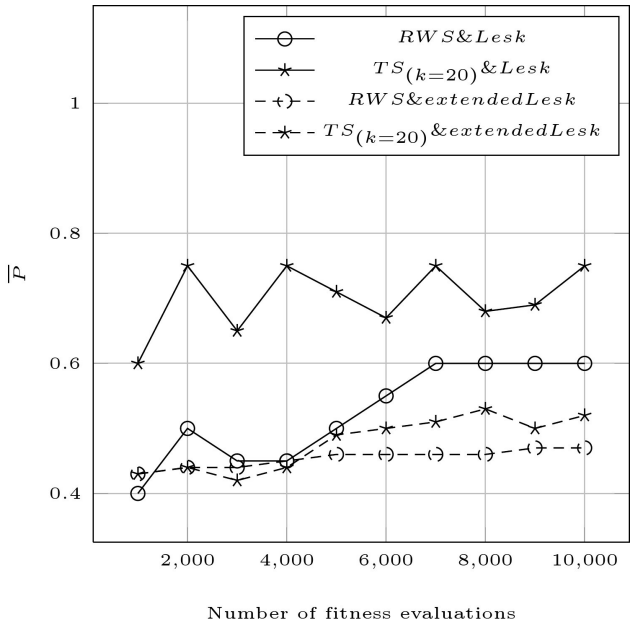


Figure 4: Mean precision of the algorithm GAWSD as a function of the number of fitness evaluations based on the parent selection method (RWS or $TS_{k=20}$) and relatedness measure ($Lesk$ or $extendedLesk$).

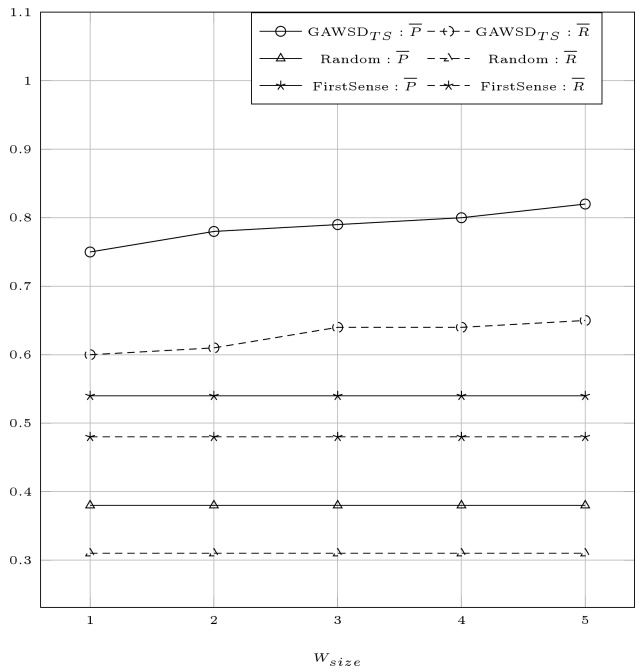


Figure 5: Mean precision and mean recall of the algorithms GAWSD$_{TS}$, Random, and FirstSense as functions of the window size.

ferent corpora. For that reason, we implemented a naïve Bayes classifier to compare its results against those given by GAWSD$_{TS}$. Table 1 presents the average results obtained by GAWSD$_{TS}$, naïve Bayes (NB), Random, and FirstSense algorithms on our corpus. The results show that the best mean precision is given by GAWSD$_{TS}$ ($\overline{P} = 0.79$). Moreover, the standard deviation of the precision ($\sigma(P) = 0.08$) demonstrates its relative robustness. Although, the best mean recall is obtained by the naïve Bayes classifier ($\overline{R} = 0.68$, $\sigma(R) = 0.24$), the mean recall of GAWSD$_{TS}$ is not significantly different ($\overline{R} = 0.63$, $\sigma(R) = 0.29$). This means that GAWSD$_{TS}$ is not only able to find more relevant word senses than the naïve Bayes classifier, but can return most relevant ones as well.

7 Discussion

We evaluated the performance of different variants of GAWSD on a set of 5218 words extracted from an Arabic corpus. The obtained results show that GAWSD$_{TS}$ is the best performing algorithm.

The results of GAWSD$_{TS}$ consistently exhibited superior performance compared with a naïve Bayes classifier and baseline algorithms. Much better precision and recall were obtained by other methods for more specific WSD tasks in Arabic, such as finding correct sense of query translation terms [6], and disambiguating polysemous and homograph Arabic nouns [27].

The results obtained with GAWSD$_{TS}$ in Arabic corroborate those obtained in previous studies on GAs for WSD in Spanish [35] and English [22,84], even though they are not comparable. They confirm that GAs represent a promising approach to WSD and particularly suitable for WSD in Arabic. Indeed, GWSD [84] evaluated on SemCor (English corpora) [58] achieved a mean recall of 71.96%, and GAMBL [22] evaluated on Senseval-3 English all-words task achieved a mean accuracy of 65.2%.

The GA component of the algorithm GAWSD is language-independent. Therefore, GAWSD can be easily adapted to solve WSD in other languages by using specific preprocessing and dictionary, given that a text in the target language can be transformed into a bag of words.

8 Conclusion and future work

This study shows that only few research work has been conducted on WSD problem in Arabic. Indeed, many successful methods have not been investigated yet for Arabic language, comparatively to other natural languages.

We have proposed an evolutionary approach to the WSD problem and applied it to an Arabic corpus. Several variants of the algorithm GAWSD were formulated and examined experimentally on a large set of words extracted from an Arabic corpus. They were assessed on the task of identifying AWN word senses, attaining 79% precision and 63% recall for GAWSD$_{TS}$. Our experiments showed that GAWSD$_{TS}$ outperformed a naïve Bayes classifier in terms of mean precision, which means that GAWSD$_{TS}$ found more relevant word senses than the naïve Bayes classifier. However, their performances in terms of mean recall were comparable, with a small advantage to the naïve Bayes classifier.

Finally, this study opens other directions for future work. The tuning of the parameters remains a major issue to optimize the performance of the proposed algorithms. The results obtained can be improved by implementing a self-adaptive GAWSD that adjusts its parameters during runtime. Furthermore, examining other methods for tuning selection pressure, and thereby exploration/exploitation tradeoff of the algorithms, can have a positive impact on the performance of GAWSD. Another important avenue of research is a thorough study of memetic algorithms for WSD, since they have outperformed GAs on several hard optimization problems.

Acknowledgements

The author gratefully thanks the anonymous referees for their constructive comments on the manuscript.

References

[1] S. Abney, and M. Light (1999). Hiding a semantic class hierarchy in a markov model. In *Proceedings of the ACL Workshop on Unsupervised Learning in Natural Language Processing*, pages 1–8.

[2] E. Agirre, and D. Martinez (2001). Learning class-to-class selectional preferences. In *Proceedings of the 2001 workshop on Computational Natural Language Learning - Volume 7*, ConLL '01, pages 3:1–3:8, Stroudsburg, PA, USA.

[3] E. Agirre, and A. Soroa (2009). Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 33–41, Stroudsburg, PA, USA.

[4] H. Al-Serhan, R. Al-Shalabi, and G. Kannan (2003). New approach for extracting arabic roots. In *Proceedings of the 2003 Arab conference on Information Technology*, ACIT'2003, pages 42–59.

[5] R. Al-Shalabi, G. Kanaan, M. Yaseen, B. Al-Sarayreh, and N. Al-Naji (2009). Arabic query expansion using interactive word sense disambiguation. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

[6] F. Amed, and A. Nürnberger (2008). Arabic/english word translation disambiguation using parallel corpora and matching schemes. In *Proceedings of the 12th Annual Conference of the European Association for Machine Translation*, EAMT 2008, pages 6–11.

[7] L. Araujo (2008). Evolutionary parsing for a probabilistic context free grammar. In *Revised Papers from the Second International Conference on Rough Sets and Current Trends in Computing*, RSCTC '00, pages 590–597, London, UK, Springer-Verlag.

[8] L. Araujo (2002). Part-of-speech tagging with evolutionary algorithms. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 230–239, London, UK, Springer-Verlag.

[9] L. Araujo (2007). How evolutionary algorithms are applied to statistical natural language processing. *Artif. Intell. Rev.*, 28:275–303.

[10] M. Attia (2008). *Handling Arabic morphological and syntactic ambiguities within the LFG framework with a view to machine translation*. PhD thesis, University of Manchester, UK.

[11] S. Banerjee, and T. Pedersen (2003). Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 805–810, San Francisco, CA, USA.

[12] M. Bin-Muqbil (2006). *Phonetic and Phonological Aspects of Arabic Emphatics and Gutturals*. PhD thesis, University of Wisconsin-Madison, WI,USA.

[13] W. Black, S. Elkateb, H. Rodriguez, M. Alkhalifa, P. Vossen, A. Pease, and C. Fellbaum (2006). Introducing the arabic wordnet project. In Fellbaum Sojka, Choi and Vossen eds, editors, *Proceedings of the Third International WordNet Conference*, pages 295–300. Masaryk University, Brno.

[14] S. Bordag (2006). Word sense induction: Triplet-based clustering and automatic evaluation. In *Proceedings of the 11th Conference of the European Chapter of the ACL*, pages 137–144.

[15] J. Brownlee (2011). *Clever Algorithms: Nature-Inspired Programming Recipes*. LuLu.

[16] T. Buckwalter (2004). Buckwalter Arabic Morphological Analyzer (BAMA) version 2.0. Linguistic Data Consortium (LDC), University of Pennsylvania, Philadelphia, PA, USA.

[17] A. Chalabi (1998). Sakhr: Arabic-english computer-aided translation system. In *Proceedings of the Third Conference of the Association for Machine Translation in the Americas on Machine Translation and the Information Soup*, AMTA '98, pages 518–521, London, UK, Springer-Verlag.

[18] M. Ciaramita, and M. Johnson (2000). Explaining away ambiguity: learning verb selectional preference with bayesian networks. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, COLING '00, pages 187–193, Stroudsburg, PA, USA.

[19] S. Clark, and D. Weir (2002). Class-based probability estimation using a semantic hierarchy. *Comput. Linguist.*, 28:187–206.

[20] M. Davis, and T. Dunning (1996). Query translation using evolutionary programming for multilingual information retrieval ii. In *Evolutionary Programming*, pages 103–112.

[21] K. De Jong (2006). *Evolutionary computation - a unified approach*. MIT Press.

[22] B. Decadt, V. Hoste, W. Daelemans, and A. den Bosch. GAMBL, genetic algorithm optimization of Memory-Based WSD. In Rada Mihalcea and Phil Edmonds, editors, *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3)*, pages 108–112.

[23] M. Diab (2009). Second generation tools (AMIRA 2.0): Fast and robust tokenization, pos tagging, and base phrase chunking. In Khalid Choukri and Bente Maegaard, editors, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, pages 285–288, Cairo, Egypt.

[24] M. Diab (2003). *Word Sense Disambiguation within a Multilingual Framework*. PhD thesis, University of Maryland College Park, USA.

[25] M. Diab (2004). An unsupervised approach for bootstrapping arabic sense tagging. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Semitic'04, pages 43–50, Stroudsburg, PA, USA.

[26] M. Diab, M. Alkhalifa, S. Elkateb, C. Fellbaum, A. Mansouri, and M. Palmer (2007). Semeval 2007 task 18: Arabic semantic labeling. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 93–98, Stroudsburg, PA, USA.

[27] K. Elghamry (2006). Sense and homograph disambiguation in arabic using coordination-based semantic similarity. In *Proceedings of AUC-OXFORD Conference on Language and Linguistics*.

[28] G. Escudero, L. Màrquez, and G. Rigau (2000). Boosting applied to word sense disambiguation. In *Proceedings of the 11th European Conference on Machine Learning*, ECML '00, pages 129–141, London, UK, Springer-Verlag.

[29] G. Escudero, L. Màrquez, G. Rigau, and J. Salgado (2000). On the portability and tuning of supervised word sense disambiguation systems. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing an Very Large Corpora*, pages 172–180.

[30] A. Farghaly, and K. Shaalan (2009). Arabic natural language processing: Challenges and solutions. *ACM Trans, Asian Lang. Inform. Process.*, 8:14:1–14:22.

[31] C. Fellbaum (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.

[32] L. Fogel, A. Owens, and M. Walsh (1966). *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, USA.

[33] W. Gale, K. Church, and D. Yarowsky (2004). Using bilingual materials to develop word sense disambiguation methods. In *Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 101–112.

[34] M. Galley, and K. McKeown (2003). Improving word sense disambiguation in lexical chaining. In *Proceedings of the 18th international Joint Conference on Artificial Intelligence*, IJCAI'03, pages 1486–1488, San Francisco, CA, USA.

[35] A. Gelbukh, G. Sidorov, and S. Han (2003). Evolutionary approach to natural language word sense disambiguation through global coherence optimization. *WSEAS Transactions on Communications*, 1:11–19.

[36] T. Gharib, M. Habib, and Z. Fayed (2009). Arabic text classification using support vector machines. *International Journal of Computers and Their Applications*, 16(4):192–199.

[37] A. Gliozzo, B. Magnini, and C. Strapparava (2004). Unsupervised domain relevance estimation for word sense disambiguation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 380–387.

[38] N. Habash, and O. Rambow (2005). Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting on ACL*, ACL '05, pages 573–580, Stroudsburg, PA, USA.

[39] S. Harabagiu, G. Miller, and D. Moldovan (1999). WordNet 2 – a morphologically and semantically enhanced resource. In *Proceedings of the ACL SIGLEX Workshop: Standardizing Lexical Resources*, pages 1–8.

[40] G. Hirst, and D. St Onge (1998). Lexical chains as representation of context for the detection and correction malapropisms. In *WordNet: An Electronic Lexical Database, C. Fellbaum, Ed.*, pages 305–332. MIT Press, Cambridge, MA.

[41] J. Holland (1975). *Adaptation in natural and artificial systems*. University of Michigan press, Ann Arbor, Cambridge, MA, USA.

[42] J. Jiang, and D. Conrath (1997). Semantic similarity based on corpus statistics and lexical taxonomy. http://www.citebase.org/abstract?id=oai:arXiv.org:cmp-lg/9709008

[43] B. Keller, and R. Lutz (1997). Evolving stochastic context-free grammars from examples using a minimum description length principle. In *Workshop on Automata Induction, Grammatical Inference and Language Acquisition (ICML097)*.

[44] S. Khoja (2003). Stemmer. http://zeus.cs.pacificu.edu/shereen/research.htm#stemming

[45] J. Koza (1992). *Genetic programming*. MIT Press, Cambridge, MA, USA.

[46] M. Lapata, and F. Keller (2007). An information retrieval approach to sense ranking. In *Human Language Technologies 2007: Proceedings of the Conference of the North American Chapter of the ACL*, pages 348–355, Rochester, New York.

[47] C. Leacock, G. Miller, and M. Chodorow (1998). Using corpus statistics and wordnet relations for sense identification. *Comput. Linguist.*, 24:147–165.

[48] Y. Lee, and H. Ng (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 41–48, Stroudsburg, PA, USA.

[49] E. Lefever, V. Hoste, and M. De Cock (2011). Parasense or how to use parallel corpora for word sense disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 317–322, Stroudsburg, PA, USA.

[50] M. Lesk (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA, ACM.

[51] H. Li, and N. Abe (1998). Generalizing case frames using a thesaurus and the MDL principle. *Comput. Linguist.*, 24:217–244.

[52] D. Lin (1998). An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA.

[53] J. Mallery (1988). *Thinking about foreign policy: Finding an appropriate role for artificial intelligence computers*. PhD thesis, MIT Political Science Department, Cambridge, MA, USA.

[54] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll (2004). Finding predominant word senses in untagged text. In *Proceedings of the 42nd Annual Meeting on ACL*, ACL '04, pages 280–287, Stroudsburg, PA, USA.

[55] Z. Michalewicz (1994). *Genetic algorithms + data structures = evolution programs (2nd, extended ed.)*. Springer-Verlag New York, Inc., New York, NY, USA.

[56] R. Mihalcea (2004). Co-training and self-training for word sense disambiguation. In Hwee, editor, *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning*, pages 33–40.

[57] R. Mihalcea, P. Tarau, and E. Figa (2004). Pagerank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, pages 1126–1132, Stroudsburg, PA, USA.

[58] G. Miller, C. Leacock, R. Tengi, and R. Bunker (1993). A semantic concordance. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, pages 303–308, Stroudsburg, PA, USA.

[59] S. Mohammad, and G. Hirst (2006). Determining word sense dominance using a thesaurus. In *Proceedings of the 11th Conference on European Chapter of the ACL*, EACL, pages 121–128.

[60] R. Mooney (1996). Comparative experiments on disambiguating word senses: an illustration of the role of bias in machine learning. In *Proceedings of the 1996 Conference on Empirical Methods in Natural Language Processing*, pages 82–91.

[61] R. Navigli, and M. Lapata (2010). An experimental study of graph connectivity for unsupervised word sense disambiguation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(4):678–692.

[62] R. Navigli (2009). Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41:10:1–10:69.

[63] R. Navigli, and P. Velardi (2005). Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:1075–1086.

[64] D. Palmer (2002). Text pre-processing. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, Boca Raton, FL.

[65] P. Pantel, and D. Lin (2002). Discovering word senses from text. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 613–619, New York, NY, USA.

[66] T. Pedersen (1998). *Learning Probabilistic Models of Word Sense Disambiguation*. PhD thesis, Southern Methodist University, Dallas, TX, USA.

[67] T. Pedersen, and R. Bruce (1997). Distinguishing word senses in untagged text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*, pages 197–207, Providence, RI.

[68] R. Rada, H. Mili, E. Bicknell, and M. Blettner (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30.

[69] I. Rechenberg (1994). *Evolutionsstrategie'94*, volume 1 of *Werkstatt Bionik und Evolutionstechnik*. Friedrich Frommann Verlag, Stuttgart.

[70] P. Resnik (1993). *Selection and information: a class-based approach to lexical relationships*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA.

[71] P. Resnik (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, pages 448–453, San Francisco, CA, USA.

[72] P. Resnik (1997). Selectional preference and sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, pages 52–57.

[73] R. Roth, O. Rambow, N. Habash, M. Diab, and C. Rudin (2008). Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short'08, pages 117–120, Stroudsburg, PA, USA.

[74] M. Sawalha, and E. Atwell (2008). Comparative evaluation of arabic language morphological analysers and stemmers. In *Proceedings of 22nd International Conference on Computational Linguistics*, COLING (Posters), pages 107–110, Manchester, UK.

[75] H. Schütze (1998). Automatic word sense discrimination. *Comput. Linguist.*, 24:97–123.

[76] H-P. Schwefel (1965). *Cybernetic Evolution as Strategy for Experimental Research in Fluid Mechanics (in German)*. PhD thesis, Hermann Föttinger-Institute for Fluid Mechanics, Technical University of Berlin.

[77] M. Sussna (1993). Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the Second International Conference on Information and Knowledge Management*, CIKM '93, pages 67–74, New York, NY, USA, ACM.

[78] M. Syiam, Z. Fayed, and M. Habib (2006). An intelligent system for arabic text categorization. *International Journal of Intelligent Computing and Information Sciences*, 6(1):1–19.

[79] G. Tsatsaronis, M. Vazirgiannis, and I. Androutsopoulos (2007). Word sense disambiguation with spreading activation networks generated from thesauri. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 1725–1730, San Francisco, CA, USA.

[80] S. M. van Dongen (2000). *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, The Netherlands.

[81] Z. Wu, and M. Palmer (1994). Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 – 138, New Mexico State University, Las Cruces, New Mexico.

[82] D. Yarowsky (1994). Decision lists for lexical ambiguity resolution: application to accent restoration in spanish and french. In *Proceedings of the 32nd Annual Meeting on ACL*, ACL '94, pages 88–95, Stroudsburg, PA, USA.

[83] D. Yarowsky (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting on ACL*, ACL '95, pages 189–196, Stroudsburg, PA, USA.

[84] C. Zhang, Y. Zhou, and T. Martin (2008). Genetic word sense disambiguation algorithm. In *Proceedings of the 2008 Second International Symposium on*

Intelligent Information Technology Application - Volume 01, IITA '08, pages 123–127, Washington, DC, USA.

[85] Z. Zhong, and H.T. Ng (2010). It makes sense: a wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, ACLDemos '10, pages 78–83, Stroudsburg, PA, USA.