Ant Colony Algorithm for Arabic Word Sense Disambiguation through English lexical information

Bakhouche Abdelaali, Yamina Tlili-Guiassa, Didier Schwab, Andon Tchechmedjiev

Ant Colony Algorithm for Arabic Word Sense Disambiguation through English lexical information

Bakhouche Abdelaali*

Laboratory LRI / Team SRF
Univ. Badji Mokhtar, Annaba / Algeria
P.O. Box 12, 2300 Annaba, Algeria
E-mail: Bakhouche2006@yahoo.fr
*Corresponding author

Yamina Tlili-Guiassa

Laboratory LRI / Team SRF
Univ. Badji Mokhtar, Annaba / Algeria
P.O. Box 12, 2300 Annaba, Algeria
E-mail: guiyam@yahoo.fr

Didier Schwab

LIG (Laboratory of Informatics of Grenoble)
GETALP (Study Group for Machine Translation and Automated
Processing of Languages and Speech)
Grenoble University, France
E-mail: Didier.Schwab@imag.fr

Andon Tchechmedjiev

LIG (Laboratory of Informatics of Grenoble)
GETALP (Study Group for Machine Translation and Automated
Processing of Languages and Speech)
Grenoble University, France
E-mail: Andon.Tchechmedjiev@imag.fr

Abstract: The ability to identify the intended meanings of words in context is a central research topic in natural language. Many solutions exist for word sense disambiguation (WSD) in different languages, such as English or French, but research on Arabic WSD remains limited. The main bottleneck is the lack of resources. In this article, we show that it is possible to build a WSD system for the Arabic language thanks to the Arabic WordNet and its connexions to the English Princeton WordNet. Given that the Arabic WordNet does not contain definitions for synsets, we construct a dictionary that maps the Princeton WordNet definitions to the Arabic WordNet. We also create an Arabic evaluation corpus and gold standard. We then exploit this dictionary and evaluation corpus to run and evaluate an adapted Ant Colony algorithm on Arabic text that can use the Lesk similarity measure thanks to definition mapping. The algorithm shows a performance of approximately 80% compared to the random baseline of 78.9 %.

Keywords: Arabic language processing, Word Sense Disambiguation, Ant colony algorithm, Lesk algorithm, Local/Global algorithm, WordNet.

opinions mining, with a focus on applications in language processing and artificial intelligence, she has a number of articles in internationals journals and conferences in these subjects.

Didier Schwab received his PhD in Informatics in 2005 from the University of Montpellier, France. He was scientific fellow at University Sains Malaysia, Penang in 2006-2007. He has been assistant professor then associate professor at University of Grenoble Alpes France since 2007. He is researcher at the Study Group for Machine Translation and Automated Processing of Languages and Speech of the Laboratory of Informatics of Grenoble. His research focuses on the representation, acquisition and exploitation of meaning including Word Sense Disambiguation.

Andon Tchechmedjiev is a PhD candidate in natural language processing at Univ. Grenoble-Alpes in the GETALP research group. His research interests include multilingual lexical resources, word sense disambiguation, computational lexical semantics and machine learning. He has a Master's degree in Computer Science and A.I. from Univ. Grenoble Alpes and a BSc (Hons) First Class in Computer Science from Oxford Brookes University.

1. Introduction

Word sense disambiguation (WSD) is the process of disambiguating the sense of an ambiguous word when the word has more than one sense by using the neighbours of the word (context) in a sentence or text (Navigli, 2009; Özdemir, 2009).
Word sense disambiguation is of interest for several linguistic applications such as automatic translation, multilingual information extraction, semantic indexing, and automatic summarization.
Arabic is the fourth most-widely spoken language in the world. It is a highly inflectional language, with a rich morphology and relatively free word order. Ambiguity in Arabic is present at several levels, such as internal word structure ambiguity, syntactic ambiguity, semantic ambiguity, constituent boundary ambiguity and anaphoric ambiguity. This prevalence of ambiguity has led to difficulty with the automatic processing of Arabic (Attia, 2008).
Ant colony optimization (ACO) was first introduced by Marco Dorigo (Dorigo et al., 1996; Dorigo et al., 1999; Dorigo and Blum, 2005). This algorithm is a fairly novel technique for solving computational problems by mimicking the natural behaviour of ants as they generate and select paths to a food source from a colony. The first application of ant colony optimization was to solve the Travelling Salesman Problem and other similar np-hard combinatorial optimization problems (asymmetric traveling salesman problem, the quadratic assignment problem, and the job–shop scheduling problem). In addition, ACO has been successfully applied to transportation planning problems (Yang et al., 2007; Vitins and Axhausen, 2009). To disambiguate words in context, Schwab et al. have proposed a distinction between local and global algorithms: the similarity measure between word senses is the local measure, and the combinatorial optimization algorithm that finds the best sense combination is the global algorithm. They proposed a global ACO algorithm and compared it to Simulated Annealing and Genetic Algorithms. The ACO algorithm obtained better precision, recall

and execution time (Schwab et al., 2011; Schwab et al., 2012; Schwab et al., 2013).
In this paper, we adapt the approach of (Schwab et al., 2013) to solve the WSD problem in the Arabic language (AWSD henceforth). Firstly, our approach extracts two sets of features, namely the set of words from the Arabic WordNet and the set of meanings from the English WordNet, from which we build the dictionary. Secondly, we present the ACO algorithm used as well as the variant of the Lesk similarity measure applied.
The rest of this paper is organized as follows: in section 2, we present the specificity of the Arabic language; in section 3, we briefly discuss some of the related work in the area of Arabic Word Sense Disambiguation; in section 4, we describe the resources and algorithms used; in section 5, we present the proposed method for AWSD; and finally, we present the results of tests of our model.

2. Specificity of the Arabic language

"Arabic is the official language of hundreds of millions of people in twenty Middle East and northern African countries, and is the religious language of all Muslims of various ethnicities around the world" (Elkateb et al., 2006). Every year, Arabic written content on the web is multiplied by 250; this is the fastest growth of written content for any language on the Internet (Russian is the second fastest at 182x). There are approximately 65 million Arabic speakers on the Internet, which is 18.8% of the global Internet population (Menai et al., 2012).
The literary Arabic language is called Modern Standard Arabic MSA (Robert, 2009). It is currently the only official form of Arabic and is used in most written documents as well as in formal spoken occasions, such as lectures and news broadcasts. It is a difficult language that could hinder the development of tools for the application of natural language processing. The Arabic language has many particularities such as short vowels, the absence of capital letters and complex morphology. The Arabic language is composed of nouns, verbs and particles, wherein these

are morphemes and derived from a closed set of approximately 10,000 roots. Arabic is also highly inflectional and derivational, which makes morphological, syntactic and semantic analysis a very complex task (Farghaly and Shaalan, 2009). There is no capitalization in Arabic, which makes it hard to identify proper names, acronyms, and abbreviations.

Furthermore, the Arabic Language is highly ambiguous for several reasons.

- Vowelization is a very interesting characteristic of the Arabic language, which causes ambiguity when it is absent, and this is usually the case.
- Polysemy (when a word or phrase has several meanings) is the second problem that causes semantic ambiguity in Arabic.
- A rarity of resources such as an Arabic Corpora and pre-treatment text tools leads to a lot of time being wasted on the process of collecting and/or modelling and developing these resources.

All of these factors collectively will affect the availability of WSD applications for Arabic.

3. Previous Work

Several approaches that address the problem of Arabic word sense disambiguation have been presented in the past few years. Among the most cited is the work of Mona Diab, who used a supervised learning approach. This approach exploits translational correspondence between words in a parallel Arabic English corpus to annotate Arabic text. It exhibits an absolute precision of 56.9% on the AWSD task (Diab, 2004).

Menai and Wodjan proposed an approach to solve the Arabic word sense disambiguation problem based on a genetic algorithm. In this algorithm, a text is tokenized, and stop-words are filtered and stemmed. In the pre-processing phase, tokenization splits the text into words. After tokenization, the authors removed stop words to filter out words that were not important in the text, such as prepositions and articles. Subsequently, the authors performed stemming (Khoja's Stemmer) on the remaining tokens. In stemming, the prefixes and suffixes of the word are removed. The senses of each word were then retrieved from the Arabic WordNet (AWN).

AWSD (GA) is used to find the most appropriate mapping from words to senses retrieved from the AWN in the context T. The authors have shown that GA performs better than the Naïve Bayes algorithm. They obtained an accuracy of 78.9% (Menai et al., 2012; Menai, 2014a; Menai, 2014b).

The system presented by Elmougy, Hamza and Noman (Elmougy et al., 2008; Elmougy et al., 2012) to solve the ambiguity of non-diacritic words in the Arabic language uses the routing algorithm with a naïve Bayes classifier. It achieved a rate of precision on the order of 76.6%.

To disambiguate Arabic words, Zouaghi, Merhben and Zerigui (Zouaghi et al., 2011) tested the Naïve Bayes algorithm, the decision lists and the exemplar-based algorithm on tagged samples and a large amount of data in the used corpus. For these tests, the exemplar-based algorithms achieve rates of precision of 48.23%, 43.86% and 52.02%, respectively.

Another method for word sense disambiguation was proposed by Zouaghi, Merhben and Zerigui (Merhben et al., 2009). Their method evaluated the variants of the Lesk algorithm and used the Arabic WordNet to perform word sense disambiguation in Arabic. The modified version of the Lesk algorithm achieved a precision of 67%.

The method presented by Nameh et al. (2011) is a supervised learning method for word sense disambiguation based on the Inner Product of vectors.

The system extracts two sets of features: the set of words that have occurred frequently in the text and the set of words surrounding the ambiguous word. This approach achieves a precision of 77.1%.

Shah et al. (2010) proposed an approach for accurate prediction of Arabic text consisting of both lemmas and morphological analyses. The accuracy of their system was reported to be 90%.

In spite of the great attention the problem has received in the last years and the important developments achieved, the precision of state-of-the-art algorithms is far from being satisfactory.

4. Resources and algorithms used in our approach

In this section, we present the characteristics of the resources and algorithms used in our approach.

a. WordNet Database

A WordNet is a machine-readable lexical database that groups words into clusters of synonyms called synsets. Every synset can be thought of as representing a unique word sense (meaning or concept). A WordNet provides general definitions and examples for the synsets and includes semantic relations between them. The semantic relations allow for a WordNet to be interpreted hierarchically as a lexical ontology/taxonomy.

WordNet is like a dictionary in that it stores words and meanings. However, it differs from traditional dictionaries in many ways. For instance, words in WordNet are arranged semantically instead of alphabetically. Synonymous words are grouped together to form synonym sets, or synsets. Each such synset therefore represents a single distinct sense or concept (Miller et al., 1990; Miller 1995).

The Arabic WordNet (AWN) is a lexical database of the Arabic language following the development process of the Princeton English WordNet and the Euro WordNet. It utilizes the Suggested Upper Merged Ontology as an Interlingua to link the Arabic WordNet to previously developed WordNet[1] (Elkateb et al., 2006). The Arabic WordNet consists of 9228 synsets (6252 nominal, 2260 verbal, 606 adjectival, and 106 adverbial), containing 18,957 Arabic expressions. This number includes 1155 synsets that correspond to Named Entities, which have been extracted automatically and are being checked by lexicographers (Rodriguez et al., 2008; Elberrichi et al., 2012). These numbers are constantly changing. At the time of writing, the English WordNet consisted of 155287 synsets (117798 nominal, 11529 verbal, 21479 adjectival, and 4481 adverbial).

[1] http://www.globalwordnet.org/AWN

b. *LESK algorithm*

The Lesk algorithm is a disambiguation method based on counting the overlaps between the definitions of sense of a target word (supplied by a machine readable dictionary), and the definitions of words in the context (Lesk, 1986; Banerjee, 2002). The selected sense for this word corresponds to the definition of the sense containing the maximal number of overlaps with the context. The algorithm begins anew for each word and does not utilize the senses it previously assigned. Formally, given two words w1 and w2, the following score is computed for each pair of word senses:

$S_1 \in$ Senses (w_1) and $S_2 \in$ Senses (w_2):

Score Lesk $(S_1, S_2) = |$ gloss $(S_1) \cap$ gloss $(S_2) |$, where gloss (S_i) is the bag of words in the textual definition of sense S_i of w_i. The senses that maximize the above formula are assigned to the respective words. However, this requires the calculation of | Senses (w1) | · | Senses (w2) | gloss overlaps. The original Lesk algorithm relies on glosses found in traditional dictionaries such as the Oxford Advanced Learner's Dictionary of Current English, Collins English Dictionary and the Oxford English Dictionary (Banerjee, 2002). Another version of the Lesk algorithm takes advantage of the highly inter–connected set of relations among synonyms that WordNet offers.

c. *Ant colony algorithm (ACO)*

Ants are social insects that live in colonies that may include thousands of individuals. When the ants find food, each individual ant does not know where to go. First, ants randomly chose their path. However, their path is chosen depending on the pheromone trails that have been laid on the ground by previous ants. The trail of pheromones on the shortest path is increased faster than the longest path. In the end, all ants choose the shortest path. Three factors that allow the ants to find the shortest path are as follows:

- Ants follow pheromone trails;
- Pheromone trails degrade over time;
- Short paths take less time to traverse.

The idea of the ant colony algorithm is to mimic this behaviour with "simulated ants" walking around the graph representing the problem to be solved. The first application of ant colony optimization was to solve the Travelling Salesman Problem (Dorigo et al., 1996; Dorigo et al., 1999; Dorigo and Blum, 2005). In this well-known problem, the goal is to find, for finitely many towns whose pairwise distances are known, the shortest path connecting them. Several improvement algorithms have been used with ACO to solve such problems as the asymmetric traveling salesman problem, the quadratic assignment problem, and the job–shop scheduling problem. In addition, ACO has been applied successfully for transportation planning problems (Yang et al., 2007; Vitins and Axhausen, 2009).

d. *Local/Global algorithm*

Several word sense disambiguation systems are based on the notion of local and global algorithms (Ratinov et al., 2011; Schwab et al., 2013b). A local algorithm allows the calculation of the semantic similarity between two lexical objects (sense, words, constituents, etc.) in the textual segment. Global algorithms

propagate local measures at the upper level. In other words, this technique permits a global extension of a local algorithm based on semantic relatedness to an entire text. We use this technique in our approach: the LESK algorithm is the local algorithm, and the ant colony algorithm is the global algorithm.

5. Generation of an Arabic dictionary

External knowledge is a fundamental component of WSD. In Arabic, there are no available dictionaries in these forms. However, there has been much work done on bilingual dictionaries based on WordNet architecture (Black et al., 2006). Therefore, a bilingual dictionary has been implemented successfully for common communication words using WordNet. This source provides data that are essential to associate English senses with Arabic words (see Fig. 1). However, it has difficulty providing the Arabic equivalent word with certainty to that of an English word.
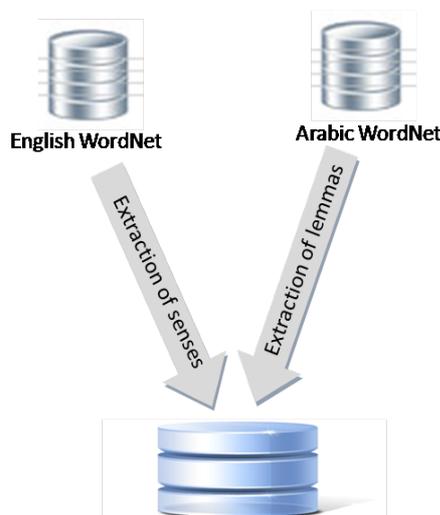

Fig. 1 Arabic Dictionary

Disambiguation of ambiguous Arabic words

The proposed work consists of developing a method used to resolve semantic ambiguity for the Arabic language using the Ant colony algorithm (see Fig. 2). In this section, we explain the most important points of this approach.

Construction of the text graph

The environment consists of a graph with ants. It is simply organized according to the elements of the text (text, sentence, word) without external language information to better understand the mechanics of the algorithms.

In our graph, we distinguish two types of nodes. Nests produce ants that move on the graph in order to find energy and bring it back to their nest. The more energy brought back by the ants, the more ants the nest can in turn produce.

Types of nodes

Nests and plain nodes are two distinct types of nodes in this graph. Following Schwab et al. (2012), each possible word sense is associated with a nest. From the point of view of an ant, a node can be any of the following:

- The ant's nest (anthill), where it was born;
- An enemy nest, which corresponds to another sense of the same word;
- A friendly nest: all of those nests that are not enemies;
- A plain node: any node that is not a nest.

Vector of definition

The odour of a nest is the numerical sense vector and corresponds to the definition of the sense associated with the nest (see Fig. 3). All ants born in the same nest have the same odour vector.

Fig. 3

```
<ids>wash#2:29:00:: </ids>
<def>0 11 11 11 11 12 12 16 16 36 36 39 39 74 82 83 83 83 92 92 98 101 101 101
```

Vector of sense's definition 'wash'

Movement of ants

An ant is born (produced by its nest) and goes in search of energy. It is drawn by nodes that carry a lot of energy and avoids arcs with too much pheromone to enable exploration of more solutions.

Bridge

When an ant arrives on a node adjacent to a potential friend nest (i.e., that corresponds to a sense of a word), it has to decide between taking any of the possible paths or going to that nest node. The only difference is that if the ant chooses to go on the potential friend nest, a bridge between that nest and the ant's home nest is built and the ant follows it to go home. Bridges behave like regular edges, except that if the concentration of pheromone on them reaches a certain level, they collapse and are removed.

Text

↓

Segementing and
Lemmatizing

↓

Arabic WSD
with ACO

↓

Result

Arabic WordNet

Aa
Dictionary

Princeton Wordnet

Fig. 2 Diagram of the proposed approach for AWSD

Interpretative paths

For all intents and purposes, bridges behave exactly like ordinary paths, however, when the concentration of pheromone on a bridge reaches zero, the bridge "evaporates" and ceases to exist.

When many bridges have been created, some will be strengthened (when many ants follow them) and other will disappear. This will lead to a monopolization of energy by the anthills linked by the strongest bridges. From the point of view of linguistics, our bridge construction model corresponds to emergent interpretative paths.

At the end of the simulation, for each word node, the anthill with the highest energy level is chosen as the solution.quality of the results.

The choice of an answer

The selected answers correspond to words in the nest node with the highest energy value. The reason for this choice over using the pheromone concentration is that empirically, the energy level correlates better with the actual F1 scores. In turn, the global LESK score of a selected sense combination has an even better correlation with the F1 score. This is why we retain the sense combinations resulting from each iteration of the algorithm (highest energy nests at each iteration) and select the one with the highest global LESK score as the final solution.

Illustrated example

The easiest way to understand how the ant colony works is by means of an example; we illustrate the steps of the algorithm through the path taken by an ant on the phrase:

"poet said rhyme". ''قال الشاعر بيتا''

We follow the path of an ant that was born in the anthill /بيت Rhyme - corresponding to the second meaning for the word, بيت.

Step 1: The ant begins to search for food to take energy to its nest. It begins to explore the graph starting with the name corresponding to the بيت# n node.

Step 2: On the path to this node, it deposits its pheromone, and arriving, it takes energy. It also deposits some of its odour on the node. This odour is marked by the deposition of two components taken at random from its mother nest.

Step 3: The ant then chooses to follow another path where it deposits its pheromone. It will also deposit two vector components of the mother nest at random and take energy. At this stage, the ant has a full quantity of energy, and a pseudo-random decision will make it return to the mother nest node.

Step 4: As an ant cannot go on a node from which it came, unless that is its only chance, our ant has the choice between going on to قال # v- or الشاعر# n.

Step 5: Ants coming from الشاعر/ Poet part of their vector on الشاعر# n have more in common with the vector node than our ants, بيت# v-.

Step 6: Our ant therefore is more likely to go to الشاعر# n. It does this while depositing its pheromone along the way. The ant leaves its odour on the destination node but does not take energy, as it is a back node. Now, its only option is to go to الشاعر/ Poet -.

Step 7: For the ant, the الشاعر/ Poet-friend is a potential node. It therefore has the opportunity to build a bridge to its mother anthill. Therefore, the ant chooses to borrow and deposit its
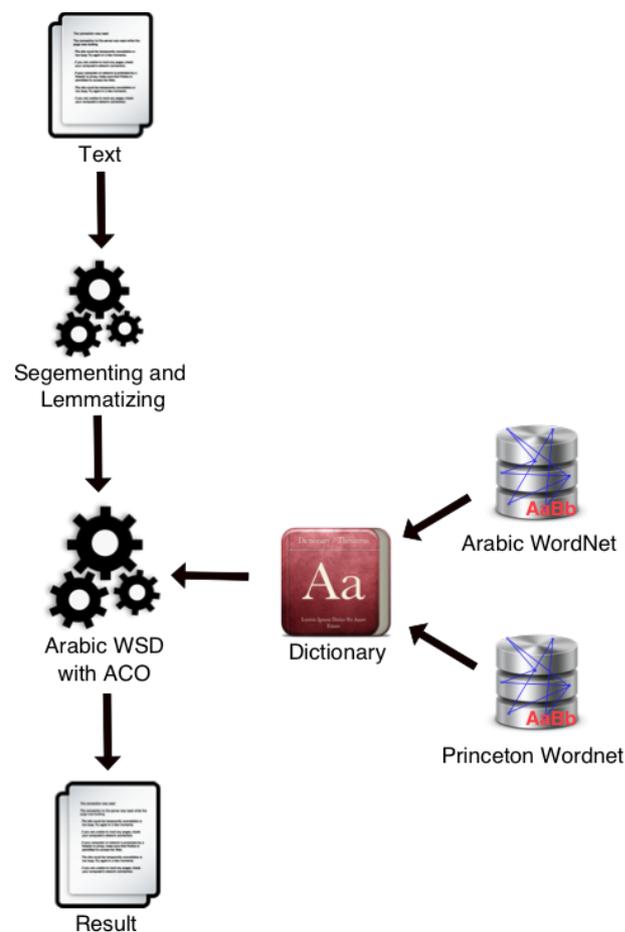
pheromone. It reaches its mother nest and deposits all of its energy. The ant then returns to search mode and resumes its travels to the node where it will die and deposit its transported energy and the energy it took to make it.
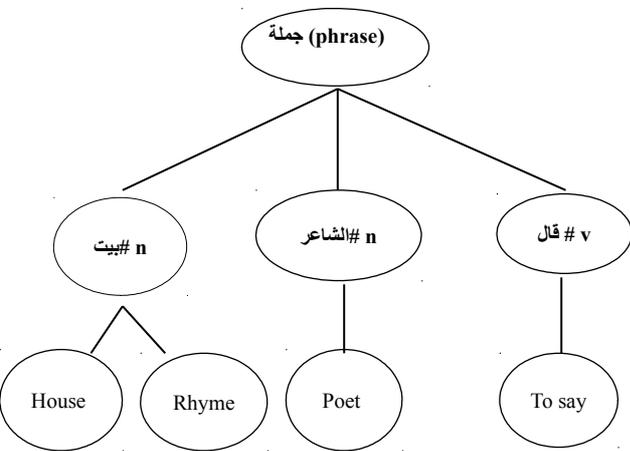

Fig. 4 Example of phrase "قال الشاعر بيتا" **"poet said rhyme "**

6. Experimental evaluation

In this section, we present the results of our experiments.

a. Statistics of the Arabic dictionary

The main statistics of the Arabic dictionary that we created are summarized in Table I. The first column of this table gives the parts of speech, while the other columns indicate the number of Arabic parts of speech and the number of English sense definitions. Fig. 5 shows the screen of an English-Arabic dictionary based on WordNet capture. We see that Arabic words do not sufficiently correspond to all possible English senses.

	Arabic POS	English Sense
Nouns	9641	14680
Verbs	2777	6084
Adjectives	662	762
collocations	12905	263
Total	25985	21789

Table 1: Statistics of the Arabic dictionary

b. Data pre-processing

The experimental dataset consists of six Arabic texts of various domains (culture, economy, international, local, religion, and sports). These texts are extracted from newspaper articles, which were recorded in the corpus of Elwatan (Abbas et al., 2011). Table 2 describes the characteristics of the corpus.

Domains	Number of articles
Culture	2782
Economy	3468
International	2035
Local	3596
Religion	3860
Sports	4550
Total	20291

Table 2: Statistics of the corpus

To study the effect of the ACO algorithm, we have conducted six groups of experiments, for each group and for each text category, one-third of the articles were randomly specified and used for testing and the remaining articles were used for training. Table 3 illustrates the domains addressed by these texts and the distribution of words as described in the texts.

Text	Domain	Words	Annotated
D001	Culture	800	242
D002	Economy	2130	1133
D003	International	600	325
D004	Local	755	432
D005	Religion	295	125
D006	Sport	450	298
TOTAL	-	5030	2555

Table 3: The six articles in the dataset

Pre-processing aims to transform the Arabic text documents into a form that is suitable for the algorithm.

- Each document in the Arabic dataset is processed to remove digits, numbers, hyphens, and punctuation marks.
- We have normalized the letters "
- All non-Arabic texts were removed.
- Arabic function words (stop words) were removed. In these experiments, we utilized the stop word list generated by the website Sourceforge[2], a list containing 26037 words.

All of the results of the algorithm were obtained using a PC with Intel Core Duo CPU 2.66 and 6.GB RAM of main memory on Windows 7. The application was developed in Java language. Figure 8 shows the screen of our applications.

c. Evaluation metrics

Evaluation criteria are attempted to indicate how many words the system can disambiguate. The precision is a measurement of how many target words are correctly disambiguated and gives the accuracy of the system.

$$Precision = \frac{¿\, correctly\ disambiguated\ words}{¿\, disambiguated\ words}$$

Recall is defined by the ratio of the number of correctly disambiguated target words to the total number of target words in the dataset.

2http://sourceforge.net/projects/arabicstopwords/

$$Recall = \frac{¿ correctly\ disambiguated\ words}{¿ tested\ set\ words}$$

The F-measure is the harmonic mean of the precision and recall values, as described in the following equation:

$$F-measure = 2.\frac{precision\ . recall}{precesion + recall}$$

For all of our experiments, we used these three evaluation measures (Recall, Precision, and F-measure) as the bases of our comparisons.

d. Selection of the parameters

The choice of parameters is critical to the performance of the algorithm. The first set of experiments involved the numerical investigation of the ACO parameters and their impact on the performance of AWSD. Table 4 shows the parameters of the ACO.

Notation	Description	Value range	Estimation for Text
GetEnerg	Energy taken by an ant when it arrives on a node	1-30	9.0
EnergMax	Maximum quantity of energy an ant can carry	1-60	22.142
pheroEvap	Evaporation rate of the pheromone between two cycles	0.0-1.0	0.3577
EnergInit	Initial quantity of energy on each node	5-60	32.0
AntLife	Ant-life-span	1-30	**27.0**
TailleVector	Odour vector length	20-200	135.637
depotVector	Percentage of the odour vector component (words) deposited by an ant when it arrives on a node	0-100	0.9775
NbrCycle	Number of cycles of the simulation	1-100	-

Table 4: Parameters of the algorithm

Notation	Value
Number of Nodes	415
Number of Paths	4096
Number of Ants	1026
Cycle Number	99

Table 5: Simulation Environment for Text d001

The algorithm does not give the same meanings on each execution; execution of the repeat is required at least twice to get the exact sense. We have noted that after several cycles (approximately 70), the results obtained are similar, as shown in Figure 4, and we can achieve the same results between two executions.
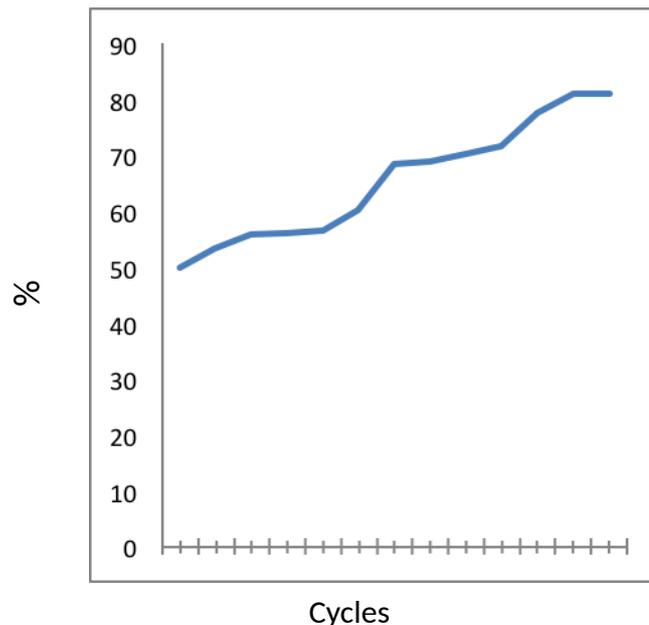

Fig. 5 The F-measure result generated by the algorithm against six Arabic datasets. In each dataset, we arbitrarily consider 70% of documents for training and 30% for testing.

e. Comparison with other methods

Several approaches that address the problem of Arabic word sense disambiguation have been presented in section 3. The result of these methods cannot be compared directly to our algorithms' results because those methods are for different tasks and their results were generated using different datasets.

For that reason, in this comparison, we apply the same experimental data that we have used before in the experimental study of the Genetic algorithm (Menai et al., 2012), Naïve Bayes classifier (Elmougy et al., 2008) and Modified version of the Lesk (Merhben et al., 2009)

Methods	Score
Ant colony algorithm (our approach)	80 %
Genetic algorithm	78.9 %
Naïve Bayes classifier	76.6 %
Modified version of the Lesk	67%

Table 6: Comparison with other methods

The results obtained with the ant colony algorithm corroborate those obtained in previous studies on ACO for WSD in English (Schwab et al., 2011, 2012, and 2013) even though they are not comparable.

Our experiments show that our approach exhibits superior results when compared with existing techniques. This is not surprising because our approach attempts to globally optimize the text cohesion, while the methods we used as a baseline optimize it locally. Much better precision and recall were obtained by other

methods for more specific WSD tasks in Arabic, such as finding the correct sense of query translation terms, and disambiguation of polysmous and homograph Arabic nouns.

We evaluated the performance of different variants of the ACO algorithm on a set of 5000 words extracted from an Arabic corpus. The results obtained show that ACO is the best performing algorithm.

7. Conclusion and future work

In this paper, we presented an approach of word sense disambiguation for the Arabic language that tackled several difficulties, in particular the rareness of studies in the field of semantics offering integrated solutions and the lack of lexical resources, in addition to technical problems related to programming imposed by the nature of the Arabic language.

To achieve this, we set our objectives at the beginning and we organized our work into three main steps. First, we conducted a theoretical study on the field. We collected and prepared all of the necessary linguistic data: a body of work and a lexical database. Then, we developed our system. Finally, we concluded with a quantitative and qualitative evaluation of our system.

To the best of our knowledge, there is no published research proposing the Ant colony algorithm-based approach to solve the AWSD problem. The ACO algorithm has the advantage of being easily adapted to a dynamic environment, which is important for the WSD problem because natural languages can evolve quickly.

There is still a lot of work to be done to introduce the Arabic language to the world of NLP, with wider opportunities than is the case with other languages. We believe and hope that this study offers solutions that can be exploited in the developing use of the Arabic language in the world of information technology, communications and electronic management.

```
<word tag="‫ﻞﺴﻏ‬%v">
<sense>
<ids>wash%2:29:00:: </ids>
<def>0 11 11 11 11 12 12 16 16 36 36 39 39 74 82 83 83 83 92 92 98 101 101 101 155 244 244 244 309 404 404 429 447 458 623 623 762 828 939 1077 115
</sense>
<sense>
<ids>clean%2:30:01:: </ids>
<def>0 0 0 0 0 9 9 9 9 12 12 16 16 16 45 45 82 82 83 124 249 293 426 623 1077 1561 1572 2003 2562 3027 3027 3027 3027 3027 3027 3258 3489 3685 3989
</sense>
<sense>
<ids>wash%2:35:00:: </ids>
<def>0 0 0 0 2 2 9 9 11 11 11 11 11 11 11 11 12 12 12 12 12 16 16 16 26 26 26 36 36 36 37 37 37 39 39 82 82 82 83 83 83 92 124 124 138 146 249 321 421
</sense>
<sense>
<ids>rinse%2:35:00:: </ids>
<def>0 0 11 11 11 12 12 16 16 39 82 124 249 447 1354 2321 2562 2630 3614 5585 5599 6442 6721 7810 7810 9860 10866 11876 11876 13032 15283 15285 1
</sense>
<sense>
<ids>clean%2:35:00:: </ids>
<def>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 9 9 9 11 11 11 11 11 11 11 12 12 12 12 12 12 12 16 16 16 16 22 36 37 39 39 39 71 82 83 83 83 83 83 9
</sense>
</word>
```

Fig. 6 Part of the Arabic Dictionary

```xml
<?xml version="1.0" encoding="utf-8" ?>
<corpus lang="ar">
<text id="d001">
<sentence id="d001.s001">
مع
<instance id="d001.s001.t001" lemma="تزايد" pos="n">تزايد</instance>
<instance id="d001.s001.t002" lemma="حاجة" pos="n">الحاجة</instance>
إلي
<instance id="d001.s001.t003" lemma="رشد" pos="v">ترشيد</instance>
الأنفاق ،
و
<instance id="d001.s001.t004" lemma="تعظيم" pos="n">تعظيم</instance>
<instance id="d001.s001.t005" lemma="استفادة" pos="n">الاستفادة</instance>
من
<instance id="d001.s001.t006" lemma="مورد" pos="n">الموارد</instance>
<instance id="d001.s001.t007" lemma="أتاح" pos="v">المتاحة</instance>
،
<instance id="d001.s001.t008" lemma="يتم" pos="v">يتم</instance>
<instance id="d001.s001.t009" lemma="إجراء" pos="n">إجراء</instance>
<instance id="d001.s001.t010" lemma="عديد" pos="a">العديد</instance>
من
<instance id="d001.s001.t011" lemma="مراجعة" pos="n">المراجعات</instance>
و
<instance id="d001.s001.t012" lemma="دراسة" pos="n">الدراسات</instance>
التي
<instance id="d001.s001.t013" lemma="كشف" pos="v">تكشف</instance>
عن
<instance id="d001.s001.t014" lemma="تكلفة" pos="n">تكاليف</instance>
<instance id="d001.s001.t015" lemma="نمط" pos="n">أنماط</instance>
<instance id="d001.s001.t016" lemma="تنمية" pos="n">التنمية</instance>
و
<instance id="d001.s001.t017" lemma="رقامية" pos="n">الرقامية</instance>
```

Fig. 7 Sample of graph text

```
d001  d001.s001.t001  augmentation%1:04:00::  !!  lemma=تزايد#n
d001  d001.s001.t002  need%1:26:00::  !!  lemma=حاجة#n
d001  d001.s001.t003  adulthood%1:28:00::  !!  lemma=رشد#n
d001  d001.s001.t004  idealization%1:04:00::  !!  lemma=تعظيم#n
d001  d001.s001.t005  benefit%1:21:00::  !!  lemma=الاستفادة#n
d001  d001.s001.t006  resource%1:21:00::  !!  lemma=مورد#n
d001  d001.s001.t007  leave%2:42:01::  !!  lemma=أتاح#v
d001  d001.s001.t008  orphan%2:40:00::  !!  lemma=يتم#v
d001  d001.s001.t009  proceeding%1:04:00::  !!  lemma=إجراء#n
d001  d001.s001.t010  many%3:00:00::  !!  lemma=عديد#a
d001  d001.s001.t011  follow-up%1:04:00::  !!  lemma=مراجعة#n
d001  d001.s001.t012  report%1:10:03::  !!  lemma=دراسة#n
d001  d001.s001.t013  uncover%2:39:00::  !!  lemma=كشف#v
d001  d001.s001.t014  charge%1:21:02::  !!  lemma=تكلفة#n
d001  d001.s001.t015  practice%1:04:00::  !!  lemma=نمط#n
d001  d001.s001.t016  development%1:22:02::  !!  lemma=تنمية#n
d001  d001.s001.t017  comfort%1:26:00::  !!  lemma=رقامية#n
d001  d001.s001.t018  circulate%2:38:03::  !!  lemma=أشمع#v
d001  d001.s001.t019  interstice%1:08:00::  !!  lemma=خدر#n
d001  d001.s001.t020  contract%1:10:00::  !!  lemma=عقد#n
d001  d001.s001.t021  three%1:23:00::  !!  lemma=ثلاثة#n
d001  d001.s001.t022  past%1:28:01::  !!  lemma=الماضي#n
d001  d001.s001.t001  augmentation%1:04:00::  !!  lemma=تزايد#n
d001  d001.s001.t002  need%1:26:00::  !!  lemma=حاجة#n
d001  d001.s001.t003  adulthood%1:28:00::  !!  lemma=رشد#n
d001  d001.s001.t004  idealization%1:04:00::  !!  lemma=تعظيم#n
d001  d001.s001.t005  benefit%1:21:00::  !!  lemma=الاستفادة#n
d001  d001.s001.t006  resource%1:21:00::  !!  lemma=مورد#n
d001  d001.s001.t007  leave%2:42:01::  !!  lemma=أتاح#v
d001  d001.s001.t008  orphan%2:40:00::  !!  lemma=يتم#v
d001  d001.s001.t009  proceeding%1:04:00::  !!  lemma=إجراء#n
d001  d001.s001.t010  many%3:00:00::  !!  lemma=عديد#a
d001  d001.s001.t011  check%1:04:05::  !!  lemma=مراجعة#n
```

Fig. 8. The output annotation in the format required for the evaluation script.

References

Abbas, M, Smaili, K, and Berkanis, D. (2011) "Evaluation of Topic Identification Methods on Arabic Corpora". *Journal of Digital Information Management, 2011, vol. 9, no 5.*

Al-Shalabi R., Kanaan G., Yaseen M., Al-Sarayreh B., Al-Naji N. A, (2009). "Arabic query expansion using interactive word sense disambiguation," *Proc. 2nd Int. Conf. on Arabic Language Resources and Tools, The MEDAR Consortium, Cairo, Egypt, 2009.*

Attia M. A., (2008) "Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation", a *thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Humanities*

Banerjee S. and Pedersen T. (2002), "An adapted Lesk algorithm for word sense disambiguation using WordNet". In: *Computational linguistics and intelligent text processing. Springer Berlin Heidelberg, 2002. (pp. 136-145).*

Basil J. Vitins and Kay W. Axhausen. (2009), "Optimization of Large Transport Networks Using the Ant Colony Heuristic" *Vol. 24, Issue 1, (pp. 1–14), January 2009.*

Beseiso M., Ahmad A. and Ismail R., (2010) "A Survey of Arabic Language Support in Semantic Web" *Int. J. of Computer Applications Vol. 9– No.1, Nov. 2010*

Black, William, ELKATEB, Sabri, Rodriguez, Horacio, et al. (2006) "Introducing the Arabic WordNet project". *In: Proceedings of the 3rd International WordNet Conference (GWC-06). 2006. (pp. 295-299).*

Diab M. T., (2004), "An unsupervised approach for bootstrapping Arabic sense tagging," *Proc. Work. On Computational Approaches to Arabic Script-based Languages, Semitic'04, ACL, Stroudsburg, PA, USA, (pp. 43-50), 2004.*

Dorigo M., Maniezzo V. and Colorni A., (1996) "Ant system: optimization by a colony of cooperating agents" *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on Vol. 26 N° 1 (pp. 29-41), Feb 1996.*

Dorigo M., Caro G., Gambardella L., (1999) "Ant algorithms for discrete optimization" *Artificial life Vol.5 N° 2,(pp.137-172), April 1999.*

Dorigo M., Blum C., (2005) *"*Ant colony optimization theory: A survey" *Theoretical computer science Vol. 344 N° 2 pp. 243-278, Elsevier, Nov. 2005.*

Dorigo, Marco, Birattari, Mauro, and Stutzle, Thomas. (2006)"Ant colony optimization». *Computational Intelligence Magazine, IEEE, 2006, Vol. 1, No 4, (pp. 28-39).*

Elberrichi Z. and Abidi K. (2012), "Arabic text categorization: a comparative study of different representation modes". *Int. Arab J. Inf. Technol.*, vol. 9, *N° 5,* (pp.465-470).

Elghamry K., (2006) "Sense and homograph disambiguation in Arabic using coordination-based semantic similarity" *Proc. AUC-OXFORDConference on Language and Linguistics, 2006.*

Elkateb, Sabry, BLACK and al. (2006). "Arabic WordNet and the challenges of Arabic". *In: Proceedings of Arabic NLP/MT Conference, London, UK. 2006.*

Elmougy S., Hamza T. and Noaman H. M. (2008)"Naïve Bayes classifier for Arabic word sense disambiguation "*Proc. 6th Int. Conf. on Informatics and Systems.*

Elmougy S., Hamza T. and Noaman H. M. (2012), "Lexical Disambiguation of Arabic Language: An Experimental Study" *Polibits No.46 México Jul. /dic. 2012.*

Farag A., Nürnberger A.,(2008) "Arabic/English word translation disambiguation using parallel corpora and matching schemes," *Proc.12th Annual Conf. of the European Assoc. for Machine Translation, EAMT 2008, (pp. 6-11).*

Farghaly Ali and Shaalan Khaled, (2009) "Arabic Natural Language Processing: Challenges and Solutions" *ACM Transactions on Asian Language Information Processing, Vol. 8, No. 4, Article 14, Dec 2009.*

Habash N., Rambow O., (2005) "Arabic tokenization, Part-Of-Speech tagging and morphological disambiguation in one fell swoop," *Proc.43rd Annual Meeting on ACL, ACL '05, ACL, Stroudsburg, PA, USA, 2005, (pp. 573-580).*

Lesk M. (*1986*), "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," *Proc. 5th annual int. Conf. on Systems Documentation, SIGDOC '86, ACM, New York, NY, USA, 1986, (pp. 24-26).*

Menai M. B. and Alsaeedan W., (2012) "Genetic algorithm for Arabic word sense disambiguation" *Proc. Work. 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Kyoto, Japan, August 8-10, 2012.*

Menai M. B., (2014a), "Word Sense Disambiguation using an evolutionary approach", *Informatica Vol. 38, (pp. 155–169), 2014.*

Menai M. B., (2014b), "Word sense disambiguation using evolutionary algorithms – Application to Arabic language" *Computers in Human Behavior, Vol. 41, (pp. 92–103), Elsevier, Dec 2014.*

Merhben L., Zouaghi A. and Zrigui M (2009). "Ambiguous *Arabic Words Disambiguation: The results" Student Research Workshop, RANLP 2009 - Borovets, Bulgaria, (pp. 45–52).*

Miller, George A., Beckwith, Richard, Fellbaum, Christiane, (1990),*"*Introduction to WordNet: An on-line lexical database", *International journal of lexicography, 1990, Vol. 3, No 4, (pp. 235-244).*

Miller, George A. (1995) "WordNet: a lexical database for English". *Communications of the ACM*, 1995, Vol. 38, No 11, (pp. 39-41).

Navigli, Roberto. (2009),"Word sense disambiguation: A survey". *ACM Computing Surveys (CSUR), 2009, vol. 41, no 2, pp. 10.*

Nameh M., Fakhrahmad S.M., Z. Jahromi M. (2011) "A New Approach to Word Sense Disambiguation Based on Context Similarity*" Proc. Work. Of the World Congress on Engineering, London, U.K, July 6 - 8, 2011.*

ÖzdemirV, (2009) "word sense disambiguation for Turkish lexical sample», *a thesis, July 2009, Fatih University*

Ratinov, Lev, Roth, Dan, Downey, Doug, and al. (2011) "Local and global algorithms for disambiguation to Wikipedia". *In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011.pp. 1375-1384.*

Robert A. Cote (2009), "choosing one dialect for the Arabic speaking world: a status planning dilemma" *Arizona Working Papers in SLA &Teaching, Vol. 16, (pp. 75- 97), 2009.*

Rodriguez, Horacio, Farwell, David, Farreres, Javi, (2008)*.* "Arabic WordNet: Current state and future extensions". In: *Proceedings of the Fourth Global WordNet Conference, Szeged, Hungary*. 2008.

Schwab, Didier and Guillaume, Nathan. (2011a), "A global ant colony algorithm for word sense disambiguation based on semantic relatedness". *In: Highlights in Practical Applications of Agents and Multi agent Systems. Springer Berlin Heidelberg, 2011. (pp. 257-264).*

Schwab, Didier, GOULIAN, Jérôme, Tchechmedjiev, Andon, et al. (2012). "Ant Colony Algorithm for the Unsupervised Word Sense Disambiguation of Texts: Comparison and Evaluation". *In: COLING. 2012. (pp. 2389-2404).*

Schwab Didier, Goulian, Jérôme, and Guillaume, Nathan. (2011b). "Désambiguïsation lexicale par propagation de mesures sémantiques locales par algorithmes à colonies de fourmis". *Traitement Automatique des Langues Naturelles, 2011, p. 185.*

Schwab Didier, Tchechmedjiev Andon, Goulian, Jérôme, et al (2013a). "GETALP: Propagation of a Lesk Measure through an Ant Colony Algorithm". *Atlanta, Georgia, USA, 2013, p. 232.*

Schwab Didier, Jérôme Goulian, Andon Tchechmedjiev. (2013b). "Désambiguïsation lexicale de textes: efficacité qualitative et temporelle d'un algorithme à colonies de fourmis". *TAL, Vol.54 N° 1, 2013, (pp.99-138).*

Shah R., and al. (2010). "A New Approach to Lexical Disambiguation of Arabic Text" *Proc. Work.International Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, U.S.A.2010.*

Yang Z., Yu B. and Cheng C. (2007) "A Parallel Ant Colony Algorithm for Bus Network Optimization" *Computer-Aided Civil and Infrastructure Engineering, Vol. 22, Issue 1, (pp. 44–55), January 2007.*

Zouaghi A., Merhben L., Zrigui M (2011). "Word Sense disambiguation for Arabic language using the variant of the Lesk algorithm" *Proc. Work. International conference on artificial intelligence, Las Vegas Nevada, USA, July 18-21, 2011. (pp. 561-567)*

Notes

1. http://www.globalwordnet.org/AWN
2. http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html
3. http://sourceforge.net/projects/arabicstopwords/

Figures, images and tables

Fig. 3 Vector of sense's definition 'wash'
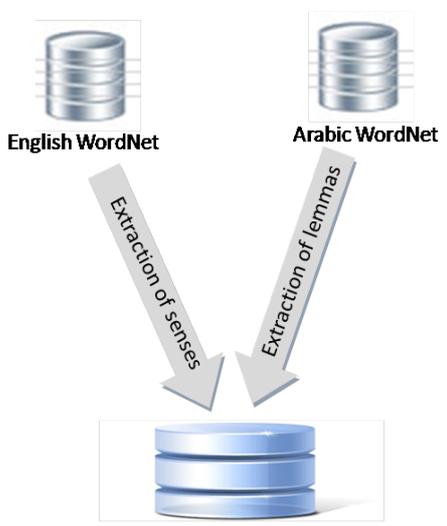

Fig. 1 Arabic Dictionary

```
(ids>wash%2:29:00:: </ids>
(def>0 11 11 11 11 12 12 16 16 36 36 39 39 74 82 83 83 83 92 92 98 101 101 101
```

Fig. 4 Example of phrase"
rhyme"

جملة (phrase)

قال الشاعر بيتا"poet said"

	Arabic POS	English Sense
Nouns	9641 # n	14680 # v
Verbs	2777	6084
Adjectives	662	762
collocations	12905	263
Total	25985	21789

Table 1: Statistics of the Arabic dictionary
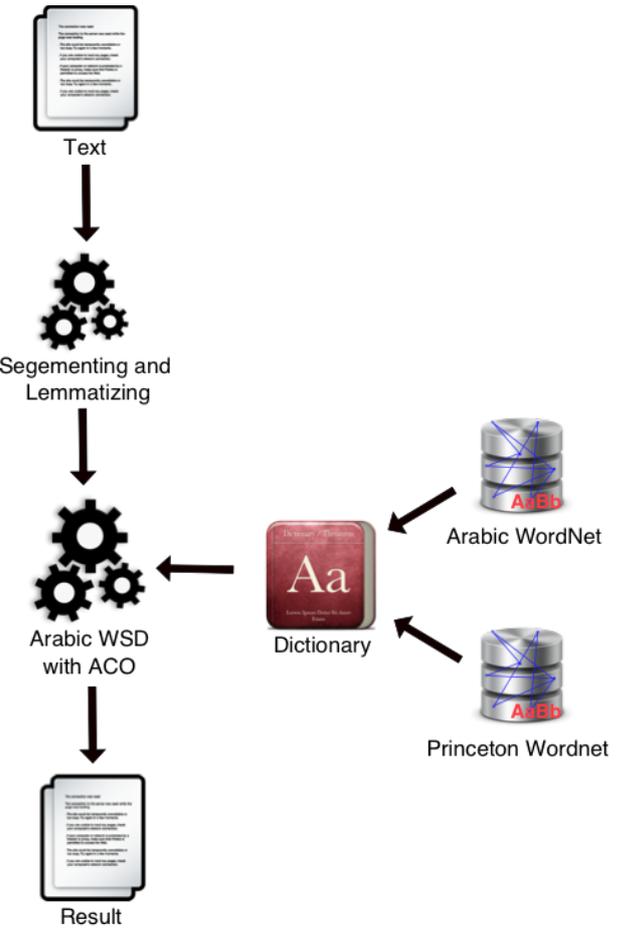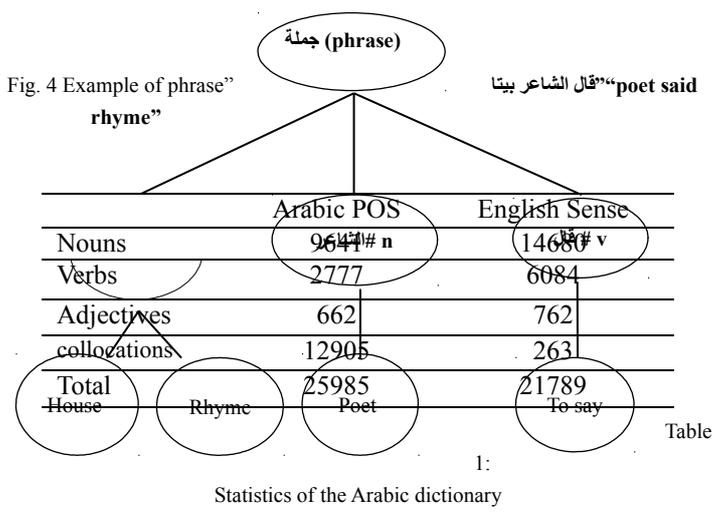
Domains	Number of articles
Culture	2782
Economy	3468
International	2035
Local	3596
Religion	3860
Sports	4550
Total	20291

Table 2: Statistics of the corpus

Text	Domain	Words	Annotated
D001	Culture	800	242
D002	Economy	2130	1133
D003	International	600	325
D004	Local	755	432
D005	Religion	295	125
D006	Sport	450	298
TOTAL	-	5030	2555

Table 3: The six articles in the dataset

Fig. 2 Diagram of the proposed approach for AWSD

Notation	Description	Value range	Estimation for Text
GetEnerg	Energy taken by an ant when it arrives on a node	1-30	9.0
EnergMax	Maximum quantity of energy an ant can carry	1-60	22.142
pheroEvap	Evaporation rate of the pheromone between two cycles	0.0-1.0	0.3577
EnergInit	Initial quantity of energy on each node	5-60	32.0
AntLife	Ant-life-span	1-30	**27.0**
TailleVector	Odour vector length	20-200	135.637
depotVector	Percentage of the odour vector component (words) deposited by an ant when it arrives on a node	0-100	0.9775
NbrCycle	Number of cycles of the simulation	1-100	-

Table 4: Parameters of the algorithm

Notation	Value
Number of Nodes	415
Number of Paths	4096
Number of Ants	1026
Cycle Number	99

Table 5: Simulation Environment for Text d001

Fig. 5 The F-measure result generated by the algorithm against six Arabic datasets. In each dataset, we arbitrarily consider 70% of documents for training and 30% for testing.

Methods	Score
Ant colony algorithm (our approach)	80 %
Genetic algorithm	78.9 %
Naïve Bayes classifier	76.6 %
Modified version of the Lesk	67%

Table 6: Comparison with other methods

Fig. 6 Part of the Arabic Dictionary

Fig. 7 Sample of graph text

```
d001  d001.s001.t001  augmentation%1:04:00::  !!  lemma=ﺗﻌﺰﻳﺰ#n
d001  d001.s001.t002  need%1:26:00::  !!  lemma=ﺣﺎﺟﺔ#n
d001  d001.s001.t003  adulthood%1:28:00::  !!  lemma=ﺭﺷﺪ#n
d001  d001.s001.t004  idealization%1:04:00::  !!  lemma=ﺗﺠﺴﻴﻢ#n
d001  d001.s001.t005  benefit%1:21:00::  !!  lemma=ﺍﺳﺘﻔﺎﺩﺓ#n
d001  d001.s001.t006  resource%1:21:00::  !!  lemma=ﻣﻮﺭﺩ#n
d001  d001.s001.t007  leave%2:42:01::  !!  lemma=ﺍﻧﺘﻘﻞ#v
d001  d001.s001.t008  orphan%2:40:00::  !!  lemma=ﻳﺘﻢ#v
d001  d001.s001.t009  proceeding%1:04:00::  !!  lemma=ﺍﺟﺮﺍﺀﺍﺕ#n
d001  d001.s001.t010  many%3:00:00::  !!  lemma=ﻋﺪﻳﺪ#a
d001  d001.s001.t011  follow-up%1:04:00::  !!  lemma=ﻣﺘﺎﺑﻌﺔ#n
d001  d001.s001.t012  report%1:10:03::  !!  lemma=ﻣﺤﺎﺿﺮﺓ#n
d001  d001.s001.t013  uncover%2:39:00::  !!  lemma=ﻛﺸﻒ#v
d001  d001.s001.t014  charge%1:21:02::  !!  lemma=ﺗﻜﺎﻟﻴﻒ#n
d001  d001.s001.t015  practice%1:04:00::  !!  lemma=ﺧﻄﺔ#n
d001  d001.s001.t016  development%1:22:02::  !!  lemma=ﺗﻄﻮﻳﺮ#n
d001  d001.s001.t017  comfort%1:26:00::  !!  lemma=ﺭﺍﺣﺔ#n
d001  d001.s001.t018  circulate%2:38:03::  !!  lemma=ﺍﻃﻼﻉ#v
d001  d001.s001.t019  interstice%1:08:00::  !!  lemma=ﻓﺠﻮﺓ#n
d001  d001.s001.t020  contract%1:10:00::  !!  lemma=ﻋﻘﺪ#n
d001  d001.s001.t021  three%1:23:00::  !!  lemma=ﺛﻼﺛﺔ#n
d001  d001.s001.t022  past%1:28:01::  !!  lemma=ﺍﻟﻤﺎﺿﻲ#n
d001  d001.s001.t001  augmentation%1:04:00::  !!  lemma=ﺗﻌﺰﻳﺰ#n
d001  d001.s001.t002  need%1:26:00::  !!  lemma=ﺣﺎﺟﺔ#n
d001  d001.s001.t003  adulthood%1:28:00::  !!  lemma=ﺭﺷﺪ#n
d001  d001.s001.t004  idealization%1:04:00::  !!  lemma=ﺗﺠﺴﻴﻢ#n
d001  d001.s001.t005  benefit%1:21:00::  !!  lemma=ﺍﺳﺘﻔﺎﺩﺓ#n
d001  d001.s001.t006  resource%1:21:00::  !!  lemma=ﻣﻮﺭﺩ#n
d001  d001.s001.t007  leave%2:42:01::  !!  lemma=ﺍﻧﺘﻘﻞ#v
d001  d001.s001.t008  orphan%2:40:00::  !!  lemma=ﻳﺘﻢ#v
d001  d001.s001.t009  proceeding%1:04:00::  !!  lemma=ﺍﺟﺮﺍﺀﺍﺕ#n
d001  d001.s001.t010  many%3:00:00::  !!  lemma=ﻋﺪﻳﺪ#a
d001  d001.s001.t011  check%1:04:05::  !!  lemma=ﻣﺘﺎﺑﻌﺔ#n
```

Fig. 8. The output annotation in the format required for the evaluation script