Swarm and Evolutionary Computation 27 (2016) 188–195

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Regular Paper

D-Bees: A novel method inspired by bee colony optimization for solving word sense disambiguation

Sallam Abualhaija *, Karl-Heinz Zimmermann

Institute of Embedded Systems, Hamburg University of Technology, Am Schwarzenberg-Campus 3 (E), 21073 Hamburg, Germany

ARTICLE INFO

Article history:
Received 7 August 2015
Received in revised form
13 November 2015
Accepted 21 December 2015
Available online 9 January 2016

Keywords:
Word sense disambiguation
Bee colony optimization
Semantic relatedness
Lesk algorithm
Metaheuristics
Text understanding

ABSTRACT

Word sense disambiguation is an early problem in the field of computational linguistics, and is defined as identifying the sense (or senses) that most likely represents a word, or a sequence of words in a given context. Word sense disambiguation was recently addressed as a combinatorial optimization problem in which the goal is to find a sequence of senses that maximizes the semantic relatedness among the target words. In this paper, we propose a novel algorithm for solving the word sense disambiguation problem, namely D-Bees, that is inspired by the bee colony optimization meta-heuristic in which several artificial bee agents collaborate to solve the problem. The D-Bees algorithm is evaluated on a standard SemEval 2007 task 7 coarse-grained English all-words corpus and is compared to the genetic and simulated annealing algorithms as well as an ant colony algorithm. It will follow that the bee and ant colony optimization approaches perform on par achieving better results than the genetic and simulated annealing algorithms on the given dataset.

1. Introduction

Word sense disambiguation (WSD) is an important problem in the field of computational linguistics since the very beginning. It is defined as identifying computationally the intended sense of a word (or a sequence of words) that is activated in a certain context [1]. For example, in the sentence "*I bought a new wireless mouse for my Apple Mac laptop*", the system should be able to select the sense of a computer device for the word *mouse* based on the provided context and similarly the computer company sense for the word *apple*. Nowadays, a massive amount of text is being widely available, so the need of obtaining efficient text-understanding systems puts the WSD task into the spot light. However, WSD is not a standalone task, rather it is being used implicitly by other applications, such as machine translation [2] or information retrieval [3].

It is worth mentioning that in this research work the task of named entity recognition (NER) is not handled. NER deals with disambiguating named entities, such as people, organizations and places in a text. For this, there are usually different evaluation datasets which are not considered here.

There exist several methods to solve WSD: supervised and unsupervised methods. *Supervised* methods apply machine learning techniques to train the classifiers on large manually-annotated corpora so that these classifiers can be used to assign new word occurrences to their most likely senses. On the contrary, *unsupervised* methods employ unannotated corpora to discriminate among word meanings based on the assumption that words which occur in similar contexts are likely to have similar meanings. More information can be found in [1].

The supervised methods achieve generally better results than the unsupervised ones [4]. However, the process of creating annotated corpora does not only need strenuous effort but is also required for every language, each sense inventory and each domain. In addition, the same language evolves by time which means even more effort to get new examples if new terms appear; e.g., the word *rock* nowadays means "a stone" or "a music genre" [5]. Therefore, their coverage depends on the set of words for which sufficient amount of examples is available.

To avoid being entrapped in the problem of creating annotated corpora, it is beneficial to do further research on unsupervised approaches. This paper focuses on unsupervised *knowledge-based* methods which rely on dictionaries and lexical knowledge resources such as WordNet [6]. The knowledge-based approaches are applicable to any text, since the knowledge resources are becoming more informative and increasingly available.

The Lesk method [7] is a well-known knowledge-based method that compares two words' definitions and calculates a score of contextual overlap between these definitions. For example, looking at the senses' definitions of the words *pine* and *cone*, there are two senses each of which includes the terms *evergreen tree*; thus

* Corresponding author.
E-mail address: sallam.abualhaija@tu-harburg.de (S. Abualhaija).

http://dx.doi.org/10.1016/j.swevo.2015.12.002
2210-6502/

these two senses will be assumed to disambiguate each other in case *pine* and *cone* co-occur in the same context.

The Lesk algorithm is applied locally for senses between two words; thus it is called a *local algorithm*. This algorithm has a main disadvantage: If there is no overlap detected between two definitions, then it will fail to disambiguate the words. This has been solved in the extended Lesk algorithm (extended Lesk) [8], which extends the two definitions by considering the definitions of the semantic related senses. We will be using this algorithm as the local algorithm in our work. Moreover, the Lesk algorithm does not take into account the assigned senses to the other words in the context. Banerjee and Pedersen [8] proposes a *global algorithm* which disambiguates all the words in a context window of size n by considering all the possible combinations of senses, each of which receives a score based on the overlap among senses' definitions along with the semantically related senses.

Pedersen et al. [9] use a brute force (exhaustive search) method to use the extended Lesk algorithm (and other local algorithms) on more than two words. Given a text and a target word, by comparing all senses of the target word with all senses of all other words in the context, the sense that yields the maximum similarity with the other words in the context is returned. However, even rather simple sentences might lead to *combinatorial explosion* especially if the words are highly polysemous which is often the case, because the relatedness function is calculated pairwise. Hence, the brute force method is not practical. Finding the exact solution of the WSD problem is a difficult task for machines, because most words have several meanings varied with the context in which they occur. The WSD problem is NP-complete [1]. This is the reason why approximation methods are being explored.

In this paper, we propose D-Bees, a novel global method inspired by bee colony optimization (BCO). D-Bees solves the WSD optimization problem for a large text efficiently by propagating the local algorithm, that is, a kind of generalization of the Lesk algorithm to a whole sentence. Our motivation is to disambiguate words as accurate as possible aiming at improving the text understanding applications, while exploring as little of the search space as possible. D-Bees can be applied to any text and is language-independent. We compare the performance of D-Bees to existing approximation algorithms, namely simulated annealing [10], genetic algorithms [11], and ant colony optimization [12].

In the following sections, first we discuss briefly the state of the art, then we give an overview of the BCO meta-heuristic in general. In Section 3 we describe the D-Bees method in details and how to adapt the principles of BCO to the WSD problem. Then we discuss the experimental settings and the results obtained. Finally, we compare the results with existing methods in Section 4. A pseudo code of the D-Bees algorithm is given in Appendix A.

2. Background knowledge

2.1. WSD as an optimization problem

We use the definition proposed by Pedersen et al. [9] to tackle WSD problem as a combinatorial optimization problem. To this end, let $C = \{w_0, w_1, w_2, \ldots, w_{n-1}\}$ be a set of n words given by a window of context of length n and w_0 be the target word to be disambiguated. Suppose each word w_i, $0 \le i \le n-1$, has m_i possible senses $s_{i1}, s_{i2}, \ldots, s_{im_i}$. Then the objective function is

$$\underset{i=1}{\overset{m_0}{\operatorname{argmax}}} \sum_{j=1}^{n-1} \max\{\text{rel}(s_{0i}, s_{j1}), \ldots, \text{rel}(s_{0i}, s_{jm_j})\}, \quad (1)$$

where rel is the semantic relatedness value between two senses.

Using the formula above, each sense of the target word is assigned a score based on the maximum relatedness with the other senses of the other words in a specific context window of size n. This refers to the brute force method proposed by Pedersen et al. as explained in the introduction. Unlike [9], we seek a sequence of senses which maximizes the overall relatedness value among the words in a given sentence. Thus, instead of assigning a score to each sense of the target word, we assign a score to a sequence of senses of all the words in the context window. This sequence of senses represents a configuration that is modified during a certain number of iterations in the global algorithm in order to find the sequence of senses that has the maximum score. A more concrete definition will be given in Section 3. In fact, this approach has the advantage that all the words in the context window are disambiguated simultaneously.

Pedersen et al. [9] have tested their brute force solution by different categories of similarity measures and different context window sizes. A variant of the Lesk measure [8] has scored overall better results.

The variant of Lesk was proposed by Banerjee and Pedersen [8] mainly to overcome the Lesk disadvantage, namely the lack of overlapping terms in the definitions of the senses. In this way, the definitions of the senses are extended to include the definitions of the semantic related senses, like the is-a relation provided by WordNet [6]. Therefore, we are encouraged to use a similar measure in our experiments, beside the fact that unlike other measures, the Lesk measure does not have preconditions regarding the part of speech of the words. Banerjee and Pedersen have modified the score of the original Lesk algorithm by taking the square of the longest sequence of one or more consecutive words that occur in both definitions [8]. Similarly, we consider the semantic related senses to extend the definitions. However, we follow Schwab et al. [13] by considering the merely bag-of-words overlap. Hence, the comparison between our method and that of Schwab et al. is more reliable.

2.2. Bee colony optimization

Bee colony optimization belongs to the swarm intelligence field. Swarm based systems are inspired by the social insects colonies, like ants, bees, wasps and termites. Social insects are thousands of individuals that collaborate by exchanging information directly or indirectly. Thus, they move beyond a limited-knowledge individual towards a collective intelligence [14] and achieve a total benefit for the sake of the colony. So, they may become able to solve complex problems. Such systems work well in unknown or highly dynamic environments and their behavior is characterized by being self-organized, autonomous, and decentralized [15]. These characteristics make them appealing to be used by complex optimization problems.

In fact, combinatorial optimization algorithms can be either constructive or improving. Constructive methods start from scratch and construct the solution step by step, like ant colony optimization. On the other hand, improving algorithms start with an initial solution using some heuristics and try to enhance it in several iterations, like simulated annealing, and genetic algorithms. BCO has an advantage that it can be both [15].

In nature, bee scouts explore initially the unknown environment looking for food resources from which they can collect nectar for the hive. Once they find a food resource, they head back to the hive and perform a dance on a dancing floor. There are two types of bee dances: a round dance indicates that the food source is close to the hive and a waggle dance is used if the food resource is further away. During the dance, the bees convey information about the direction, the distance to the food resource and the goodness of it. That is, the bee scouts actually make an