REVISTA SIGNOS. ESTUDIOS DE LINGÜÍSTICA ISSN 0718-0934
© 2017 PUCV, Chile • DOI: 10.4067/S0718-09342017000200287 • 50(94) 287-308

An associative method for Lesk-based word sense disambiguation*

Un método asociativo para la desambiguación de sentidos de palabras basado en Lesk

Sulema Torres-Ramos
UNIVERSIDAD DE GUADALAJARA
MÉXICO
sulema.torres@cucei.udg.mx

Israel Román-Godínez
UNIVERSIDAD DE GUADALAJARA
MÉXICO
israel.godinez@cucei.udg.mx

E. Gerardo Mendizabal-Ruiz
UNIVERSIDAD DE GUADALAJARA
MÉXICO
gerardo.mendizabal@cucei-udg.mx

Recibido: 07-VII-2015 / **Aceptado:** 28-X-2016

Abstract

One of the most important current problems in natural language processing is word sense disambiguation (WSD). WSD consists of identifying the correct sense of the words in a given text. In this work, we present a novel method for automatic WSD based on the simplified-Lesk algorithm. The proposed method employs Alpha-Beta associative memories for the relatedness computation between the senses of the ambiguous words and its context. The performance of this method was evaluated in terms of precision, recall, and F-score, using the semantically annotated corpora Senseval-2, Semcor, and Semeval-2007. The results show the advantages of the proposed method compared with other Lesk-based state-of-the-art methods.

Key Words: Computational linguistics, word sense disambiguation, simplified-Lesk algorithm, associative memories, Alpha-Beta associative memories.

Resumen

Uno de los problemas más importantes en procesamiento de lenguaje natural es la desambiguación de sentidos de palabras, que consiste en identificar el sentido correcto de las palabras en un texto dado. En este trabajo presentamos un método novedoso para la desambiguación automática de sentidos de palabras basado en el algoritmo simplificado de Lesk. El método propuesto utiliza las memorias asociativas Alfa-Beta para calcular la relación que existe entre los sentidos de las palabras ambiguas y su contexto. El desempeño de este método fue evaluado en términos de *precision*, *recall* y *F-score,* utilizando los corpus semánticamente etiquetados Senseval-2, Semcor y Semeval-2007. Los resultados muestran las ventajas del método propuesto en comparación con otros métodos del estado del arte basados en Lesk.

Palabras Clave: Lingüística computacional, desambiguación de sentidos de palabras, algoritmo simplificado de Lesk, memorias asociativas, memorias asociativas Alfa-Beta.

INTRODUCTION

Human language is based on the use of discrete units (i.e., words) that interact in non-random ways to construct a large variety of sentences (Ferrer i Cancho & Solé, 2001). Typically, in any language, there are many words that can have more than one meaning, generating ambiguity that can only be resolved by analyzing the context of where the word occurs. In Computational Linguistics, Word Sense Disambiguation (WSD) is one of the most important and challenging current problems. WSD refers to the "ability to computationally determine which sense of a word is adequate depending on its use in a particular context" (Navigli, 2009: 3). WSD is considered to be the most important problem to solve in automated text understanding. It is, therefore, a crucial resource for applications such as machine translation (Vickrey, Biewald, Teyssier & Koller, 2005; Carpuat & Wu, 2007; Chan, Ng & Chiang, 2007), information retrieval (Zhong & Ng, 2012), information extraction (Ciaramita & Altun, 2006), parsing (Agirre, Bengoetxea, Gojenola & Nivre, 2011; Agirre, Baldwin & Martinez 2008) and others.

In general, the various WSD methods require two steps: (i) the determination of all the possible meanings for every relevant word in the text, and (ii) the selection of the correct sense of the words in a given context. Since there are already many electronic databases and resources that can efficiently provide the different meanings of words in a given language (i.e., digital treasuries, machine readable dictionaries, ontologies, corpora and others), the majority of the efforts to solve WSD are directed developing methods that allow the automatic selection of the most proper sense of the ambiguous words in a text. These methods can be classified into four groups: supervised, unsupervised, semi-supervised, and knowledge-based approaches (Borah, Talukdar & Baruah, 2014; Nandanwar & Mamulkar, 2015).

A knowledge-based method that is widely known for WSD (Viveros-Jiménez, Gelbukh & Sidorov, 2013) is the simplified Lesk algorithm, which measures the overlap between definitions of the words (glosses) and words in the contexts (Kilgarriff & Rosenzweig, 2000). Due to fact that the overlap is considered as string matching, one of the critical challenges in determining the correct sense of a word in a sentence is that words can exist in several different forms. For example, the word 'beauty' is a noun, 'beautiful' is an adjective, 'beautify' is a verb, and 'beautifully' is an adverb. One option to overcome this challenge is to reduce the inflected or derived words to their base, root or word stem (i.e., stemming). However, the generation of an accurate stemming algorithm is another significant challenge in Natural Language Processing (Ramasubramanian & Ramya, 2013).

An alternative approach to overcome the stemming process includes considering the inflected or derived form of words as altered versions of its root form, then, it would be possible to use a pattern recognition algorithm that recognizes the root of a word, even if it is in a derivative or inflective form. A pattern recognition algorithm that can address this problem is the associative memory.

Associative memories have been an active field of scientific research in recent decades (Kohonen, 2012). The most important characteristic of an associative memory is the ability to recall output patterns from possibly modified input patterns.

Various associative memories have been reported as reliable alternatives to solve diverse research problems such as feature selection (Aldape-Pérez, Yáñez-Márquez, Camacho-Nieto & Ferreira-Santiago, 2013), medical data classification (Uriarte-Arcia, López-Yáñez & Yáñez-Márquez, 2014), gray-scale image encryption (Acevedo, Martínez, Acevedo & Yáñez, 2014), and sparse coding (Palm, 2013).

In particular, the Alpha-Beta associative memories have proven their applicability in many research areas because of their high recall power despite the simplicity of their algorithms (Acevedo-Mosqueda, Yáñez-Márquez & López-Yáñez, 2006; Román-Godínez & Yáñez-Márquez, 2007; Yáñez-Márquez, Cruz-Meza, Sánchez-Garfias & López-Yáñez, 2007; Román-Godínez, López-Yánez & Yánez-Márquez, 2009; Argüelles, Yáñez, López & Camacho, 2011; Román-Godínez, 2011; López-Yáñez, Sheremetov & Yáñez-Márquez, 2014).

In this paper, we present a knowledge-based computational method for WSD inspired by the simplified-Lesk algorithm. The proposed method (hereinafter known as ABWSD) employs an Alpha-Beta hetero-associative memory model to find the correct sense of an ambiguous word by comparing the word patterns of the glosses to the context words. This advantageous because it is possible to compare altered versions of the words, without the need for a 'stemming' procedure. The performance of the ABWSD method was evaluated by computing the precision, recall, and F-score employing the semantically annotated corpora Senseval-2, Semcor, and Semeval-2007.

The results show the advantages of the ABWSD method compared with other Lesk-based state-of-the-art methods.

The remainder of the paper is organized as follows: in Section 1, the theoretical framework regarding WSD and Alpha-Beta hetero-associative memories is presented. Section 2 describes the proposed method for word sense disambiguation. Sections 3 and 4 show the experimental results and discussion, respectively. Finally, conclusions are outlined.

1. Theoretical framework

1.1. Word Sense Disambiguation

Current approaches for automatic WSD can be classified into four groups according to the methodology employed for selecting the correct sense of the word to be disambiguated: supervised, unsupervised, semi-supervised, and knowledge-based approaches (Borah et al., 2014; Nandanwar & Mamulkar, 2015).

Supervised methods employ machine learning techniques to create classification models based on a training set of texts. The models are trained using a hand-labeled corpus. These labels indicate the correct sense of each instance or ambiguous word in its corresponding context. Examples of WSD methods in this category approach include the use of decision lists (Rivest, 1987), decision trees (Quinlan, 2014), Bayesian classifiers (Escudero, Márquez & Rigau, 2000), neural networks (Cottrell, 1989), support vector machines (Lee & Ng, 2002), maximum entropy (Tratz, Sanfilippo, Gregory, Chappell, Posse & Whitney, 2007), weighted KNN (Rezapour, Fakhrahmad & Sadreddini, 2011), and semantic diffusion kernels (Wang, Rao & Hu, 2014).

Unsupervised approaches are based on the fact that words with similar senses will have similar surrounding words and, therefore, do not rely on extensive prior training like the supervised methods. In general, unsupervised WSD methods instead of providing sense labels generate clusters of word occurrences. Examples of unsupervised methods include the use of context clustering (Ide, Erjavec & Tufis, 2001), word clustering (Bordag, 2006), graph-based methods (Sinha & Mihalcea, 2007), and Markov random fields (Chaplot, Bhattacharyya & Paranjape, 2015).

Semi-supervised approaches attempt to achieve WSD by progressively constructing classification models using a small number of hand-labeled samples as seeds (Pham, Ng & Lee, 2005). Examples of the use of this approach include the use of co-training and self-training (Mihalcea, 2004), alternating structure optimization (Ando, 2006), word embeddings in general and specific domains (Taghipour & Ng, 2015), statistical learning (Huang, Chen & Shi, 2013), and neural language models (Yuan, Doherty, Richardson, Evans & Altendorf, 2016).

Knowledge-based methods rely on the extensive use of knowledge sources (i.e., collocations, thesauri, dictionaries) to assign a sense to an ambiguous word by comparing each of its possible senses with those of the words in the context, and computing a semantic similarity metric of the relatedness of these senses. The similarity can be calculated by counting word overlaps between definitions of the words (Lesk, 1986), finding distances between concepts (Patwardhan, Banerjee & Pedersen, 2003), random walks (Agirre, de Lacalle & Soroa, 2014), and graph-based techniques (Navigli & Lapata, 2010).

In general, supervised methods perform better than the other approaches. However, their main limitation is the requirement of a vast number of manually classified examples to construct the classification model. Semi-supervised methods may require smaller training sets compared to the supervised methods. However, their performance depends on whether the samples selected as seeds can generate a classification model that is accurate to classify words that are not similar to those used as seeds. Unsupervised methods overcome the limitation of the need for a large number of training samples. However, their performance is reduced compared with the supervised methods. Knowledge-based methods also have the advantage of not requiring prior training, but in contrast to unsupervised methods, knowledge-based approaches recently have shown that, in the presence of enough knowledge or within a knowledge-based domain, these systems can out-perform supervised approaches, while providing at the same time much wider coverage (Navigli, 2012).

1.1.1. Lesk algorithm

A knowledge-based algorithm that has "inspired a whole family of methods that exploit the number of common words in two sense definitions" (Basile, Caputo & Semeraro, 2014.) is the Lesk algorithm (Lesk, 1986), which is based on the assumption that words in a given 'neighborhood' (section of text) will tend to share a common topic. This algorithm uses a dictionary of definitions (glosses) to assign a sense to an ambiguous word in a given text. The corresponding sense is selected by determining which gloss of the ambiguous word shares the largest number of words with the glosses of the context (i.e., the words surrounding the ambiguous word).

1.1.1.1. Simplified-Lesk algorithm

The main disadvantage of the Lesk algorithm is its exponential complexity (i.e. the number of comparisons increases combinatorially as the number of words to disambiguate in the text). To address this problem, a simplified version of this algorithm was proposed, where the sense of the ambiguous word is selected by determining which gloss of it contains the largest number of contextual words (Kilgarriff & Rosenzweig, 2000), see Figure 1.

Unfortunately, the process of finding the occurrences of words in the definitions or within the context is not trivial because of the possible inflectional forms of many words. Therefore, it is necessary to employ strategies such as measuring the similarity between words (i.e., word distance metrics) or perform 'stemming', which is the process of reducing inflectional forms and sometimes derivationally related forms of a word to a common base form (Gaidhane, Gondhale & Talole, 2015), so they can be analyzed as a single item. However, this process may increase the computational complexity of the WSD problem.

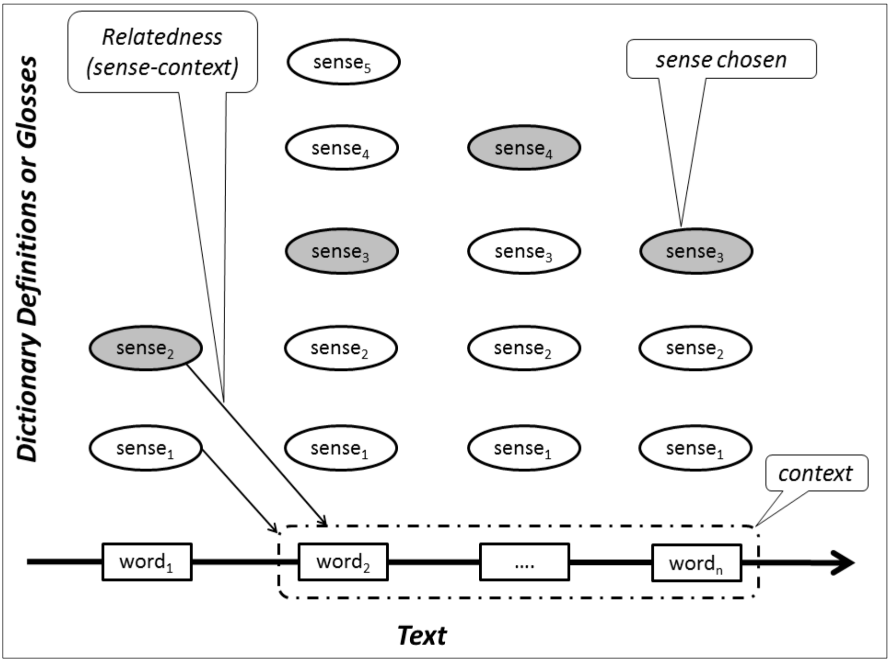

Figure 1. Graphic representation of the simplified-Lesk algorithm.

1.2. Alpha-Beta hetero-associative memories

An associative memory can be formulated as a system *M,* which relates input (x) and output (y) patterns as ($x \rightarrow M \rightarrow y$). The design of an associative memory involves two phases: (i) the learning phase on which associations between pairs of pattern vectors are defined (i.e., (x, y)) and (ii) the recall phase on which a pattern vector is presented to the associative memory to obtain the related learned pattern vector.

An associative memory is represented by a matrix generated from a finite set of pairwise pattern vector associations referred to as the fundamental set, which is described as:

$$\{ (x^{\mu}, y^{\mu}) \mid \mu = 1,2, \dots, p \} \quad (1)$$

where *p* is a positive integer representing the cardinality of the fundamental set. If for every association in the fundamental set, the input pattern is equal to the output pattern, the resulting memory is said to be auto-associative, otherwise, the memory is said to be hetero-associative. In the learning phase, an associative memory matrix *M* is generated by applying set of operators to the input and output pattern vectors. In the recall phase, another set of operators is employed to recover the related pattern vector.

Alpha-Beta hetero-associative memories presented in (Román-Godínez & Yáñez-Márquez, 2007; Román-Godínez et al., 2009), unlike original and many other models (Yáñez & Díaz de León, 2003; Ritter, Sussner & Díaz-de-León, 1998), guarantee the correct recall of the fundamental set. Alpha-beta associative memories employ the maximum $\vee$ and minimum $\wedge$, along with two binary operators: alpha (*α*) and beta (*β*), which are defined in Table 1 (Yáñez & Díaz de León, 2003). Depending on the operators employed during the learning phase, these memories can be of two types: max and min.

Table 1. Definition of the alpha and beta operators.

x_i	y_i	$\alpha(x_i, y_i)$
0	0	1
0	1	0
1	0	2
1	1	1

x_i	y_i	$\beta(x_i, y_i)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

The first step of the learning phase consists of building an output vector, y^{μ} of length *p*, for each input vector, x^{μ} of length *z*, to be recovered later depending on the type of Alpha-Beta hetero-associative memory to be computed. The output vectors would be built employing the one-hot codification (Equation 2) for the type max or the zero-hot codification (Equation 3) for type min (Román-Godínez & Yáñez-Márquez, 2007):

$$y_i^{\mu} = \begin{cases} 1, & if \quad i = \mu \\ 0, & otherwise \end{cases} \tag{2}$$

$$y_i^{\mu} = \begin{cases} 0, & if \quad i = \mu \\ 1, & otherwise \end{cases} \tag{3}$$

where $\forall i \in \{1,2,\dots,p\}$.

A matrix A^{μ} is then generated for each association(x^{μ}, y^{μ}), by employing the ⊠ operator defined as follows:

$$\mathrm{A}^{\mu}_{\mathrm{i,j}} = [y^{\mu} \boxtimes (x^{\mu})^{t}] = \alpha\left(y^{\mu}_{i}, x^{\mu}_{j}\right), \forall i \in \{1,2,\ldots,p\} \text{ and } \forall j \in \{1,2,\ldots,z\} \tag{4}$$

On the one hand, for Alpha-Beta hetero-associative memories of type max, the second step consists of applying the maximum operator ∨ to each of the 'p' matrices A^{μ}, to form the memory matrix V in the following way:

$$V^{\mu}_{i,j} = \bigvee_{\mu=1}^{p} A^{\mu}_{i,j}, \forall i \in \{1,2,\ldots,p\}\ and\ \forall j \in \{1,2,\ldots,z\} \tag{5}$$

On the other hand, for Alpha-Beta hetero-associative memories of type min, the second step consists of applying the minimum operator ∧ to each of the 'p' matrices A^{μ}, to form the memory matrix Λ in the following way:

$$\Lambda^{\mu}_{i,j} = \bigwedge_{\mu=1}^{p} A^{\mu}_{i,j}, \forall i \in \{1,2,\ldots,p\}\ and\ \forall j \in \{1,2,\ldots,z\} \tag{6}$$

1.1.3. Recall phase

In the recall phase, an input vector x^{ω} is presented to either hetero-associative memory type max (V) or min (Λ), then a vector z^{ω} is generated by applying the $\boldsymbol{\Delta_{\beta}}$ or $\boldsymbol{\nabla_{\beta}}$ operator, respectively.

For Alpha-Beta hetero-associative memories of type max, z^{ω} is obtained by:

$$z^{\omega}_{i} = V\ \Delta_{\beta}\ x^{\mu} = \bigwedge_{j=1}^{z} \beta\left(V_{i,j}, x^{\omega}_{j}\right) \tag{7}$$

For Alpha-Beta hetero-associative memories of type min, z^{ω} is obtained by:

$$z^{\omega}_{i} = \Lambda\ \mathit{\Delta}_{\beta}\ x^{\mu} = \bigvee_{j=1}^{z} \beta\left(\Lambda_{i,j}, x^{\omega}_{j}\right) \tag{8}$$

Once the z^{ω} vector has been built, it is necessary to compute a max sum vector **s** for the type max or min sum vector 'r' for the type min, with 's' and 'r' of length 'p' and theirs elements taken from the positive integer domain.

For the Alpha-Beta hetero-associative memory of type max, the 's' vector is built as follows:

$$s_i = \sum_{j=1}^{n} T_j \tag{9}$$

where $T \in \{0,1\}^n$ and its components are defined as:

$$\mathrm{T}_i = \begin{cases} 1 \leftrightarrow V_{ij} = 1 \\ 0 \leftrightarrow V_{ij} \neq 1 \end{cases} \tag{10}$$

where $\forall j \in \{1,2,\dots,n\}$.

Then, the corresponding y^{ω} vector is given as:

$$y_i^{\omega} = \begin{cases} 1 \quad if \left(s_i = \bigvee_{k \in \theta} s_k \right) \wedge (z_i^{\omega} = 1) \\ 0 \qquad\qquad otherwise \end{cases} \tag{11}$$

where $\theta = \{i | z_i^{\omega} = 1\}$.

On the other hand, for the Alpha-Beta hetero-associative memory of type min, the 'r' vector is obtained by:

$$r_i = \sum_{j=1}^{n} T_j \tag{12}$$

where $T \in \{0,1\}^n$ and its components are defined as:

$$\mathrm{T}_i = \begin{cases} 1 \leftrightarrow \Lambda_{ij} = 0 \\ 0 \leftrightarrow \Lambda_{ij} \neq 0 \end{cases} \tag{13}$$

where $\forall j \in \{1,2,\dots,n\}$.

Finally, the corresponding y^{ω} vector is given as:

$$y_i^{\omega} = \begin{cases} 0 \quad if \left(r_i = \bigwedge_{k \in \theta} s_k \right) \wedge (z_i^{\omega} = 0) \\ 1 \qquad\qquad otherwise \end{cases} \tag{14}$$

where $\theta = \{i | z_i^{\omega} = 1\}$.

2. ABWSD method

The fundamental purpose of an associative memory is to recall output patterns from possible altered input patterns (i.e., additive, subtractive or, mixed alterations). This characteristic is convenient under the scope of WSD considering that in a text, several inflectional and derivative forms of a word are found. Therefore, we propose to employ alpha-beta associative memories as a method for evaluating the similarity score between the senses of the ambiguous words and the context words.

The ABWSD consists of six steps:

1. Isolation of the ambiguous word: the ambiguous word to be evaluated is removed from the sentence; the remaining words are a considered as the 'context' set.

2. Senses acquisition: the senses (glosses) for the ambiguous word are obtained using a dictionary. Each content word (i.e., adjective, verb, noun, adverb) contained in each of the various possible senses are considered candidates to be compared with the context words as in the simplified-Lesk algorithm.

3. Binary codification: since the Alpha-Beta hetero-associative model requires binary domain vectors for its operations, the words of senses and context are mapped to a binary representation. In this case, we arbitrarily choose the ASCII code as the binary representation. However, any other binary codification could be used.

4. Vector normalization: given that the number of binary digits needed for the codification of each word may be different, the lengths of all codified word vectors are normalized to the largest length. The nonexistent components of each vector (context and sense vectors) are filled with a numeric value depending on the Alpha-Beta hetero-associative memory used, 0's for max type and 1's for min type.

5. Generation of the associative memory matrices**:** several fundamental sets are built, one per sense, according to one of the memory models presented in Section 1.2.1. For each fundamental set, a learning matrix is then generated.

6. Selection of the sense: each vector of the context words is presented to each of the learning matrices and the recall process is executed. This results in an output vector which determines the relatedness between a word of the context and a sense. Finally, for all output vectors related to each learning matrix, the sum of all components equal to 1 or 0 (depending on the type of memory) is computed. Since, each learning matrix represents a sense of the ambiguous word; the learning matrix which scores the largest sum (voting) is selected as the correct sense. If more than one sense is selected, then it is considered that the method is unable to determine the sense for the ambiguous word.

For example, consider the following sentence: "The banker deposits money in my father's accounts". To perform WSD on the word 'deposits', employing the ABWSD max type method, we perform the following steps:

1. Ambiguous word = deposits, Context = {banker, money, father, accounts}
2. S1 = {put, bank, account, banker}, S2 = {put, something, somewhere, firmly, posited, hand, shoulder, deposit, suitcase, bench, fix, eyes, spot}, S3 = {fix, force, implant, lodge, bullet, table}
   €

Table 2. Output vectors obtained per matrix and its corresponding sum of components.

Learning Matrix	Context Vector	Output Vector	Sum of Components
LMS1	c^1	$(0001)^t$	1
	c^2	$(0000)^t$	0
	c^3	$(0000)^t$	0
	c^4	$(0010)^t$	1
Total			2
LMS2	c^1	$(0000000000000)^t$	0
	c^2	$(0000010000000)^t$	1
	c^3	$(0000000000000)^t$	0
	c^4	$(0000000000000)^t$	0
Total			1
LMS3	c^1	$(000000)^t$	0
	c^2	$(000000)^t$	0
	c^3	$(000000)^t$	0
	c^4	$(000000)^t$	0
Total			0

3. Experimental results

3.1. Data sets

To evaluate the performance of the ABWSD method, we carry out WSD using three databases: (i) the semantically annotated corpus for the Senseval-2 English all-words task which consists of three documents with 2,456 words in 238 sentences (Palmer, Fellbaum, Cotton, Delfs & Dang, 2001), (ii) the English Semcor corpus (Fellbaum, 1998) created at Princeton University by the WordNet Project research team, which consists of more than 192,000 words in more than 19,000 sentences, and (iii) the semantically annotated corpus for the Semeval-2007 English all-words task which consists of 432 words in 96 sentences (Pradhan, Loper, Dligach & Palmer, 2007). The dictionary from which the senses were obtained was WordNet (Miller, 1995).

3.2. Performance measures

The performance of the ABWSD method was evaluated by computing the following measures:

- Precision is determined by the number of correct sense assignments divided by the number of words for which the method assigned a sense.
- Recall is determined by the number of correct sense assignments divided by the total of ambiguous words.
- F-Measure represents the weighted harmonic mean of precision and recall and is determined by: (2 x Precision x Recall) / (Precision + Recall)

3.3. Results

The ABWSD was implemented in two versions employing (i) associative memories type max (ABWSD$_{max}$) and (ii) associative memories type min ABWSD$_{min}$). In addition, to have a baseline the simplified-Lesk algorithm was implemented. These three methods were tested over the Senseval-2 and Semcor corpora under the following conditions: 1) with and without using stemming pre-processing in order to assess the robustness of both methods with respect to word inflections, 2) using a context window of one sentence and 3) without using an alternative strategy to choose a sense when it is not capable of assigning one.

Tables 3 and 4 list the results for ABWSD and simplified-Lesk methods for the Senseval-2 and Semcor, respectively.

In order to compare the ABWSD method with current algorithms as the ones proposed by Wang and Hirst (2014); Ramakrishnan, Prithviraj and Bhattacharyya (2004); Naskar and Bandyopadhyay (2007), the ABWSD was adjusted to work under their same conditions and corpus.

Table 3. Results over the Senseval-2 corpus for ABWSD methods and the baseline.

Method / Performance	ABWSD$_{max}$ %	ABWSD$_{min}$ %	Simplified-Lesk %
Stemming			
Precision	41.35	37.25	**53.78**
Recall	**25.43**	23.03	10.34
F-measure	**31.49**	28.46	17.35
Not stemming			
Precision	43.25	41.20	**55.93**
Recall	**23.68**	22.82	7.86
F-measure	**30.60**	29.37	13.78

Table 4. Results over the Semcor corpus for ABWSD methods and the baseline.

Method Performance	**ABWSD$_{max}$ %**	**ABWSD$_{min}$ %**	**Simplified-Lesk %**
Stemming			
Precision	36.79	39.25	**43.68**
Recall	22.80	**26.12**	9.08
F-measure	28.15	**31.37**	15.03
Not stemming			
Precision	38.45	39.91	**45.02**
Recall	21.13	**22.06**	7.14
F-measure	27.27	**28.41**	12.33

First, the ABWSD method is compared against Wang and Hirst (2014). They present a "Lesk-based algorithm which replaces the overlap mechanism of the Lesk algorithm with a general purpose Naive Bayes model" (Wang & Hirst, 2014: 1). The Naive Bayes model for word sense disambiguation (hereinafter known as NaiveBayesSM), computes the *a posteriori* probabilities of the senses of a polysemous word, then, the sense of the greater probability is chosen as the correct one. The experiments performed in Wang and Hirst (2014) use the gloss description as the information source, a one-sentence context window, and stemming of the words in glosses and context. Table 5 shows the F-score for the NaiveBayesSM and the ABWSD methods over the Senseval-2 corpus. For this comparison, the ABWSD methods were fitted with a random selection strategy for those cases when it is not able to provide an answer.

Table 5. Comparison using Senseval-2 for NaiveBayesSM and ABWSD.

Method	Gloss type	Window context	Stemming	F-score
NaiveBayesSM	Descriptive	1	Yes	36.2
ABWSD$_{min}$	Descriptive	1	Yes	41.75
ABWSD$_{max}$	Descriptive	1	Yes	**43.25**

The Gloss-centered algorithm consists of "comparing the current context for a word against a repository of contextual clues or glosses for each sense of each word" (Ramakrishnan et al., 2004: 1). With respect to the glosses or clues, three different sets of information were compiled: i) the concatenation of hypernyms names of a word-sense, ii) the concatenation of descriptive glosses of a word-sense with the glosses of its hypernyms, and iii) the concatenation of descriptive glosses of a word-sense with the glosses of its holonyms. Moreover, the context window considered for their experimentation are of one, two and three sentence sizes; and as an extra source of information, the algorithm implements a 'full content expansion' process which substitutes each word in the context with its glosses. Finally, when the algorithm is not able to choose a correct sense, the most frequent sense in the dictionary is selected.

In order to have a fair comparison with the ABWSD method, the results presented in Table 6 show those that consider: one sentence context window, no full content expansion, and no stemming. Moreover, we add to our proposal the most frequent sense.

Ramakrishnan et al. (2004) present separately the F-score for nouns and verbs. In order to compare with our proposal, the average of both scores was computed.

Table 6. Comparison using Semcor for Gloss-centered and ABWSD.

Method	Gloss type	Window context	Stemming	F-score
Gloss-centered	Hypernymy	1	No	56.45
ABWSD$_{min}$	Descriptive	1	No	**57.28**
ABWSD$_{max}$	Descriptive	1	No	56.51

Finally, we compared the ABWSD against the JU-SKNSB presented in Naskar and Bandyopadhyay (2007). JU-SKNSB uses "the meanings (i.e., glosses) of the words, as well as the words that are related to them through various relationships defined in WordNet" (Naskar & Bandyopadhyay, 2007: 204). In spite of the fact that the simple-Lesk algorithm presents a local approach for WSD, the JU-SKNSB presents a global one, turning it into a more computationally complex proposal. For the experimental purposes, they used a three-sentence window context, one before and after the target sentence. Table 7 shows the F-score obtained for JU-SKNSB and ABWSD methods over the Semeval-2007 corpus.

Table 7. Comparison using Semeval-2007 for JU-SKNSB and ABWSD.

Method	Gloss type	Window context	Stemming	F-score
JU-SKNSB	Descriptive+ relationships	3	Yes	**37.5**
ABWSD$_{min}$	Descriptive	1	Yes	13.43
ABWSD$_{max}$	Descriptive	1	Yes	11.85

4. Discussion

Note that for the baseline related experiments, the ABWSD implementations have a higher recall percentage in contrast with the simplified-Lesk. The outstanding case is presented in the Semcor corpus using stemming, where ABWSD$_{min}$ obtained 26.12% in contrast with the 9.08% obtained by the simplified-Lesk. Note that the simplified-Lesk algorithm has the higher precision. However, due to its low recall, the ABWSD implementations can be said to perform better according to the F-score. The leading case for the Senseval-2 corpus shows 30.60% for the ABWSD$_{max}$ against 13.78% for the simplified-Lesk. For the Semcor corpus, the ABWSD$_{min}$ achieves an F-score of 31.37% in contrast with the 15.03% for the simplified-Lesk. With regard to using stemming, it can be noted that when this process is not used, all the implementations suffer a reduction in their performance. However, this behavior is most noticeable in the simplified-Lesk algorithm while in the ABWSD methods this reduction is significantly less. Finally, considering that the “F-score is useful to compare systems with a coverage lower than 100%” (Navigli, 2009: 42), and this measure indicates an overall performance relating the precision and recall, it can be observed that the ABWSD shows significant improvement over the baseline.

In regards to the experimental comparison of the state-of-art, the results reported in Table 5 show that ABWSD$_{max}$ presents an F-score of 43.25 over the 36.2 reported by NaiveBayesSM, assessed over the Senseval-2 corpus. Both methods use the simplified-Lesk algorithm but replace the overlapping machinery by the Naive Bayes model and the Alpha-Beta hetero-associative memory, respectively. In order to have a fair comparison, and given that the NaiveBayesSM reports a one hundred percent coverage, the ABWSD methods were fitted with a random selection strategy for those cases when it is not able to provide an answer, keeping them into the knowledge-based approach.

Table 6 shows that ABWSD$_{min}$ is just better than the Gloss-centered algorithm with a 57.28 over 56.45, respectively. The Semcor corpus was used for this comparison. In Ramakrishnan et al., (2004) they reported several experimental results under different conditions, but the only result to which the ABWSD could be compared was the one without ‘full content expansion’, without stemming, one-sentence context window, and using the hypernyms instead of the gloss description. It is noteworthy that, due to the Gloss-centered uses a most frequent sense strategy for those cases when it is not able to provide an answer. The ABWSD was adjusted to use the same strategy. However, this adjustment is only for comparison purposes, since if a system uses “sense frequency information that is only obtainable from sense-annotated corpora, it is essentially a supervised system” (Wang & Hirst, 2014: 534).

Finally, Table 7 shows that ABWSD$_{min}$ and ABWSD$_{max}$ have a lower performance in respect to JU-SKNSB, 13.43 and 11.85 against 37.5 respectively. As we mentioned before, the JU-SKNSB has big differences, with respect to ABWSD, which improve its performance. First, according to Naskar and Bandyopadhyay (2007), it is considered a global approach instead of a local one. Second, the JU-SKNSB not only uses the glosses description of the ambiguous word but also different relations obtained from WordNet. Third, the comparisons were made using the Semeval-2007 corpus which is formed by very short sentences compared with other corpora. Having short sentences strongly affects the performance of the ABWSD given that the one and only source of information used to recall a sense for an ambiguous word is its context. Therefore, the fewer the words used to recall a sense, the lower the performance of the algorithm. In this regard, the JU-SKNSB increases the context window size and its information by using not only the context words but also their own glosses descriptions.

CONCLUSIONS

In this paper, a novel computational method for WSD based on a simplified-Lesk algorithm and Alpha-Beta hetero-associative memories with complete recall (ABWSD) was presented. The main contribution of the ABWSD method is that the use of associative memories adds robustness with respect to possible inflectional forms of the words without the need of a stemming procedure.

In the experimental comparison, it is shown that the ABWSD has better performance with respect to the baseline, because the simplified-Lesk algorithm is very sensitive to the exact string matching, so any inflectional form of the words can radically change its results.

In the comparison with the NaiveBayesSM, which is the most similar method to ABWSD, the latter exceeds the former's performance under the same conditions (context window and gloss type). It is important to note that in both cases a stemming processing is done, and even so, the ABWSD does best, demonstrating that handling inflectional form is not the only contributing factor.

Likewise, in order to compare the ABWSD method against the Gloss-centered, under similar conditions, the ABWSD was adjusted to use the most frequent sense strategy, which showed a better performance than the Gloss-centered. Nevertheless, this adjustment was made only for comparison purposes, since the use of most frequent sense strategy is considered to be a supervised methodology, which is not the intention of this work.

Finally, the JU-SKNSB presents some of the best results for the knowledge-based algorithms. However, it differs substantially from the ABWSD, in both the experimental conditions and the word sense disambiguation methodology. Therefore, the more information provided to the algorithms, the better their performance. Nonetheless, we have to be aware that the algorithms, such as JU-SKNSB, that handle large amounts of information require more computational resources. Therefore, making an effort to improve the algorithms instead of just adding information is a good alternative, and is the intention of this work.

As a future work effort, different binary mapping codifications will be used to find the one that performs better. In order to confirm the language independence, the ABWSD could be tested using different experimental sets and dictionaries. It would also be interesting to increase the context window and the glosses information without affecting the computational cost or to make a parallel implementation of the ABWSD to increase the time response.

REFERENCES

Acevedo, M. E., Martínez, J. A., Acevedo, M. A. & Yáñez, C. (2014). Morphological associative memories for gray-scale image encryption. *Appl. Math*, *8*(1), 127-134.

Acevedo-Mosqueda, M. E., Yáñez-Márquez, C. & López-Yáñez, I. (2006). Alpha-beta bidirectional associative memories based translator. *International Journal of Computer Science and Network Security*, *6*(5A), 190-194.

Agirre, E., Baldwin, T. & Martínez, D. (2008). Improving parsing and PP attachment performance with sense information. In *Proceedings of ACL-08: HLT* (pp. 317-325).

Agirre, E., Bengoetxea, K., Gojenola, K. & Nivre, J. (2011). Improving dependency parsing with semantic classes. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short papers-Volume 2* (pp. 699-703). Association for Computational Linguistics.

Agirre, E., de Lacalle, O. L. & Soroa, A. (2014). Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, *40*(1), 57-84.

Aldape-Pérez, M., Yáñez-Márquez, C., Camacho-Nieto, O. & Ferreira-Santiago, Á. (2013). Feature selection using associative memory paradigm and parallel computing. *Computación y Sistemas, 17*(1), 41-52.

Ando, R. K. (2006). Applying alternating structure optimization to word sense disambiguation. *Proceedings of the Tenth Conference on Computational Natural Language Learning* (pp. 77-84). Association for Computational Linguistics.

Argüelles, A., Yáñez, C., López, I. & Camacho, O. (2011). Prediction of CO and NOx levels in Mexico City using associative models. *Artificial Intelligence Applications and Innovations*, 364, 313-322.

Basile, P., Caputo, A. & Semeraro, G. (2014). An enhanced lesk word sense disambiguation algorithm through a distributional semantic model. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 1591-1600).

Borah, P. P., Talukdar, G. & Baruah, A. (2014). Approaches for word sense disambiguation –A survey. *International Journal of Recent Technology and Engineering*, *3*(1), 35-38.

Bordag, S. (2006). Word sense induction: Triplet-based clustering and automatic evaluation. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)* (pp. 137-144).

Carpuat, M. & Wu, D. (2007). Improving statistical machine translation using word sense disambiguation. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learnin* (pp. 61-72).

Chan, Y. S., Ng, H. T. & Chiang, D. (2007). Word sense disambiguation improves statistical machine translation. *Annual Meeting-Association for Computational Linguistics*, *45*(1), 33.

Chaplot, D. S., Bhattacharyya, P. & Paranjape, A. (2015). Unsupervised word sense disambiguation using markov random field and dependency parser. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (pp. 2217-2223).

Ciaramita, M. & Altun, Y. (2006). Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (pp. 594-602). Association for Computational Linguistics.

Cottrell, G. W. (1989). *A connectionist approach to word sense disambiguation*. Pitman: London, U.K.

Escudero, G., Márquez, L. & Rigau, G. (2000). Naive Bayes and exemplar-based approaches to word sense disambiguation revisited. *Proceedings of the 14th European Conference on Artificial Intelligence* (ECAI) (pp. 421-425), Berlin, Germany.

Fellbaum, C. (1998). *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press.

Ferrer i Cancho, R. & Solé, R. V. (2001). The small world of human language. *Proceedings of the Royal Society of London B: Biological Sciences*, 268, 2261-2265.

Gaidhane, M. S. C., Gondhale, M. D. P. & Talole, M. P. (2015). A comparative study of stemming algorithms for natural language processing. I*nternational Journal of Engineering, Education and Technology (ARDIJEET)*, *3*(2), 1-6.

Huang, Z., Chen, Y. & Shi, X. (2013). A novel word sense disambiguation algorithm based on semi-supervised statistical learning. *International Journal of Applied Mathematics and Statistics*™, *43*(13), 452-458.

Ide, N., Erjavec, T. & Tufis, D. (2001). Automatic sense tagging using parallel corpora. In Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium -*NLPRS* (pp. 83-90).

Kilgarriff, A. & Rosenzweig, J. (2000). Framework and results for English SENSEVAL. *Computers and the Humanities*, *34*(1-2), 15-48.

Kohonen, T. (2012). *Self-organization and associative memory*. Springer Berlin Heidelberg.

Lee, Y. K. & Ng, H. T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10* (pp. 41-48). Association for Computational Linguistics.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. *Proceedings of the 5th annual international conference on Systems documentation* (pp. 24-26). ACM.

López-Yáñez, I., Sheremetov, L. & Yáñez-Márquez, C. (2014). A novel associative model for time series data mining. *Pattern Recognition Letters*, 41, 23-33.

Mihalcea, R. (2004). Co-training and self-training for word sense disambiguation. *Proceedings of the 8th Conference on Natural Language Learning* (pp. 33-40).

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, *38*(11), 39-41.

Nandanwar, L. & Mamulkar, K. (2015). Supervised, semi-supervised and unsupervised WSD approaches: An overview. *International Journal of Science and Research (IJSR), 4*(2), 1684-1688.

Naskar, S. K. & Bandyopadhyay, S. (2007). JU-SKNSB: Extended WordNet based WSD on the English all-words task at SemEval-1. *Proceedings of the 4th International Workshop on Semantic Evaluations* (pp. 203-206). Association for Computational Linguistics.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR), 41*(2), 10.

Navigli, R. (2012). A quick tour of word sense disambiguation, induction and related approaches. *International Conference on Current Trends in Theory and Practice of Computer Science* (pp. 115-129). Springer Berlin Heidelberg.

Navigli, R. & Lapata, M. (2010). An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE transactions on pattern analysis and machine intelligence*, *32*(4), 678-692.

Patwardhan, S., Banerjee, S. & Pedersen, T. (2003). Using measures of semantic relatedness for word sense disambiguation. *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 241-257). Springer Berlin Heidelberg.

Palm, G. (2013). Neural associative memories and sparse coding. *Neural Networks*, 37, 165-171.

Palmer, M., Fellbaum, C., Cotton, S., Delfs, L. & Dang, H. T. (2001). English tasks: All-words and verb lexical sample. *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems* (pp. 21-24). Association for Computational Linguistics.

Pham, T. P., Ng, H. T. & Lee, W. S. (2005). Word sense disambiguation with semi-supervised learning. *Proceedings of the National Conference on Artificial Intelligence*, *20*(3), 1093.

Pradhan, S. S., Loper, E., Dligach, D. & Palmer, M. (2007). SemEval-2007 task 17: English lexical sample, SRL and all words. *Proceedings of the 4th International Workshop on Semantic Evaluations* (pp. 87-92). Association for Computational Linguistics.

Quinlan, J. R. (2014). *C4. 5: Programs for machine learning.* San Mateo, CA: USA.

Ramakrishnan, G., Prithviraj, B. & Bhattacharyya, P. (2004). A gloss-centered algorithm for disambiguation. *Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text.*

Ramasubramanian, C. & Ramya, R. (2013). Effective pre-processing activities in text mining using improved porter's stemming algorithm. *International Journal of Advanced Research in Computer and Communication Engineering, 2*(12), 2278-1021.

Rezapour, A. R., Fakhrahmad, S. M. & Sadreddini, M. H. (2011). Applying weighted KNN to word sense disambiguation. *Proceedings of the World Congress on Engineering*, 3, 6-8.

Ritter, G. X., Sussner, P. & Díaz-de-León, J. L. (1998). Morphological associative memories. *Neural Networks, IEEE Transactions on*, *9*(2), 281-293.

Rivest, R. L. (1987). Learning decision lists. *Machine learning, 2*(3), 229-246.

Román-Godínez, I. (2011). Identification of functional sequences using associative memories. *Revista Mexicana de Ingeniería Biomédica*, *32*(2), 109-118.

Román-Godínez, I. & Yáñez-Márquez, C. (2007). Complete recall on alpha-beta heteroassociative memory. In *MICAI 2007: Advances in Artificial Intelligence* (pp. 193-202). Springer Berlin Heidelberg.

Román-Godínez, I., López-Yáñez, I. & Yáñez-Márquez, C. (2009). Classifying patterns in bioinformatics databases by using Alpha-Beta associative memories. *Biomedical Data and Applications* (pp. 187-210). Springer Berlin Heidelberg.

Sinha, R. S. & Mihalcea, R. (2007). Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. *Proceedings of the First IEEE International Conference on Semantic Computing* (pp. 363-369).

Taghipour, K. & Ng, H. T. (2015). Semi-supervised word sense disambiguation using word embeddings in general and specific domains. *Proceedings of NAACL HLT* 2015, 314-323.

Tratz, S., Sanfilippo, A., Gregory, M., Chappell, A., Posse, C. & Whitney, P. (2007). PNNL: A supervised maximum entropy approach to word sense disambiguation. *Proceedings of the 4th International Workshop on Semantic Evaluations* (pp. 264-267). Association for Computational Linguistics.

Uriarte-Arcia, A. V., López-Yáñez, I. & Yáñez-Márquez, C. (2014). One-hot vector hybrid associative classifier for medical data classification. *PloS one*, *9*(4), e95715.

Vickrey, D., Biewald, L., Teyssier, M. & Koller, D. (2005). Word-sense disambiguation for machine translation. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 771-778). Association for Computational Linguistics.

Viveros-Jiménez, F., Gelbukh, A. & Sidorov, G. (2013). Simple window selection strategies for the simplified lesk algorithm for word sense disambiguation. In F. Castro & A. Gelbukh (Eds.), *Mexican International Conference on Artificial Intelligence* (pp. 217-227). Springer Berlin Heidelberg.

Wang, T. & Hirst, G. (2014). Applying a Naive Bayes Similarity Measure to Word Sense Disambiguation. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)* (pp. 531-537). Association for Computational Linguistics.

Wang, T., Rao, J. & Hu, Q. (2014). Supervised word sense disambiguation using semantic diffusion kernel. *Engineering Applications of Artificial Intelligence*, 27, 167-174.

Yáñez, M. C. & Díaz de León, S. J. L. (2003). Memorias asociativas basadas en relaciones de orden y operaciones binarias. *Computación y Sistemas*, *6*(4), 300-311.

Yáñez-Márquez, C., Cruz-Meza, M. E., Sánchez-Garfias, F. A. & López-Yáñez, I. (2007). Using alpha-beta associative memories to learn and recall rgb images. *Advances in Neural Networks–ISNN* 2007 (pp. 828-833). Springer Berlin Heidelberg.

Yuan, D., Doherty, R., Richardson, J., Evans, C. & Altendorf, E. (2016). Word sense disambiguation with neural language models. *arXiv preprint arXiv:1603.07012.*

Zhong, Z. & Ng, H. T. (2012). Word sense disambiguation improves information retrieval. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1* (pp. 273-282). Association for Computational Linguistics.

*ACKNOWLEDGMENT

Work was done under the partial support of Mexican Government (CONACYT, SNI) and PRODEP (projects 227835 and 230707).