Available online at www.sciencedirect.com

ScienceDirect

Computer Speech and Language 41 (2017) 128–145

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Improving selection of synsets from WordNet for domain-specific word sense disambiguation

Ivan Lopez-Arevalo [a,*], Victor J. Sosa-Sosa [a], Franco Rojas-Lopez [a], Edgar Tello-Leal [b]

[a] *Science and Technology Park TecnoTam, Cinvestav Tamaulipas, Victoria 87130, Mexico*
[b] *Universidad Autonoma de Tamaulipas, Matamoros 8 y 9, Victoria 87000, Mexico*

Received 5 November 2014; received in revised form 29 April 2016; accepted 21 June 2016
Available online 23 June 2016

Abstract

Word Sense Disambiguation (WSD) is a fundamental task useful for Information Retrieval, Information Extraction, web search, and indexing, among others. In the literature there exist several works dedicated to generic WSD task, but in recent years domain-specific WSD has attracted the attention of several researchers. In this sense, this paper describes an approach for domain-specific WSD by selecting the predominant sense (synset from WordNet) of ambiguous words. To achieve it the method uses two corpora: the *domain-specific test corpus* (containing target ambiguous words) and a *domain-specific auxiliary corpus* (obtained by using relevant words from the *domain-specific test corpus*). The approach has four main stages: (1) auxiliary corpus generation; (2) *related features extraction* (from the auxiliary corpus); (3) *test features extraction* (from the test corpus); and (4) *features integration*. The proposed approach has been tested on domain-specific corpora (Sports and Finance) and on one balanced corpus, BNC. Even though our WSD approach showed some limitations when dealing with the general-domain corpus, the obtained results for domain-specific corpora, which are our main interest, were better than those reported in previous works.

Keywords: Domain-specific word sense disambiguation; WordNet; Synset; Context

1. Introduction

In Computational Linguistics, word sense disambiguation (WSD) is the task to determine which sense of a word is correct in a particular context (Agirre and Edmonds, 2007). WSD has been widely tackled with supervised, unsupervised, and knowledge-based approaches (Agirre and Edmonds, 2007; McCarthy, 2009; Navigli, 2009a). Most of the works on WSD is dedicated to *general-domain WSD*, but in recent years *specific-domain WSD* has attracted the focus. The attention on specific-domain WSD is because of the limitation of use of general-domain WSD systems in new domains, because such systems must be retrained (Faralli and Navigli, 2012). A general-domain WSD system suffers a degradation of performance when is trained and tested on different corpora – from different domains (Agirre and Lopez, 2009; Escudero et al., 2000). General-domain WSD systems yield inaccurate results when the input text

* Corresponding author at: Science and Technology Park TecnoTam, Cinvestav-Tamaulipas, Victoria 87130, Mexico. Tel.: +52 834 1070258; fax: +52 834 1070224.
E-mail address: ilopez@tamps.cinvestav.mx (I. Lopez-Arevalo).

http://dx.doi.org/10.1016/j.csl.2016.06.003
0885-2308/

contains words with several meanings – depending on contexts (Navigli et al., 2011). Many systems of web search, information retrieval, and text mining, among others, apply general-domain WSD, but results even suffer ambiguity; parts of the results are different from the desired ones. Several approaches have been proposed to tackle this problem: (a) semi-supervised, supervised, and active learning approaches for domain adaptation use source domain data supplemented with unlabeled data from the target domain (Agirre and Lopez, 2008, 2009; Faralli and Navigli, 2012; Seng and Ng, 2007); (b) the predominant sense acquisition approach builds a thesaurus from raw text and then retrieves neighbor words for each ambiguous word; these words are used to find the predominant sense in texts (Koeling et al., 2005; McCarthy et al., 2004, 2007; Reddy et al., 2010; Tejada-Cárcamo et al., 2010); (c) the knowledge-based approach explores the information contained in lexical resources, such as WordNet (Miller, 1995), supplemented by unlabeled data (Agirre et al., 2009; Navigli et al., 2011).

The domain-specific WSD poses new challenges (Agirre et al., 2009), such as: (a) different domains involve several sense distributions and predominant senses; (b) some words tend to occur in fewer senses; (c) the context of senses might change and new senses and words might be involved. Practical benefits of domain-specific WSD approaches can be found on machine translation systems (Agirre and Edmonds, 2007).

Independent of general or specific-domain WSD, there are two variants of the WSD task (Mihalcea *et al.*, 2004; Snyder and Palmer, 2004; Agirre and Edmonds, 2007, Agirre et al., 2010): (a) *All words*. In this task the objective is to disambiguate all ambiguous words in a text. The system requires large knowledge about the domain, and the training corpus must be wide enough to cover the whole domain. In general, these systems are supervised or semi-supervised; they are trained by using instances manually labeled (training corpus), then these systems are applied over a set of unlabeled instances (test corpus) to find the correct sense for ambiguous words; (b) *Lexical sample*. In this task the objective is to disambiguate one ambiguous word, usually one word per sentence (from the test corpus). Systems in this task only need information for the ambiguous words, so there is more flexibility in training. A training corpus is not defined previously, and these systems use *on the fly* knowledge (which constitutes the training corpus). The manner how the "training corpus" is generated makes the difference of these WSD approaches.

This article describes an approach for domain-specific WSD on the *lexical sample task* by selecting the predominant synset[1] from WordNet for an ambiguous word. The approach is based on the extraction, representation, and integration of information provided by *features* of ambiguous words on two corpora: *auxiliary* and *test*. The proposed approach was tested on two domain-specific corpora (Sports and Finance domains) and on one general-domain corpus (British National Corpus [BNC]). The obtained results on the Sports and Finance domains outperform those reported in the literature, but not is the case on BNC.

The rest of the paper is organized as follows, Section 2 presents previous work on the domain-specific WSD, Section 3 explains the proposed approach, Section 4 describes the obtained results, Section 5 gives a discussion of results, and Section 6 presents some conclusions.

2. Domain-specific WSD

There exist several approaches discussed in the literature about domain-specific WSD (Agirre and Edmonds, 2007; McCarthy, 2009; Navigli, 2009a). Most of these approaches have good performance on disambiguate words when are trained and tested on the same corpus (in the same domain), but its performance drops when working on unrelated data to the training corpus or different domain (Escudero et al., 2000). This performance drop can mainly be attributed to the domain dependence of the WSD systems (Escudero et al., 2000; Gale et al., 1992; Martínez and Agirre, 2000). To solve this dependence it is necessary to include a dynamic adaptation step in the WSD approach when the training is carried out. Seng and Ng (2007) proposed a supervised machine learning approach to tackle this problem. They added training examples from a new domain as additional training data to a WSD system, showing that the effectiveness of the adaptation process is improved when the predominant sense of the target domain is used. A minimally-supervised approach was proposed by Faralli and Navigli (2012), they acquired a multi-domain glossary from the Web with minimal supervision; this process was carried out through web queries. The acquired definitions were used as a sense repository for domain-specific WSD. Agirre and Lopez (2009) proposed a semi-supervised approach by using Singular Value Decompositions to find correlations between words from unlabeled data to overcome the lack of training data, and consequently to get a better domain adaptation for WSD. An interesting approach was presented

[1] See the definition of *synset* in Section 3.1 *Definitions*.

by Reisinger and Mooney (2010), proposing a multi-prototype vector-space model for representing word meaning. This approach eases the disambiguation of words independently of the domain. The model represents a word's meaning by means of a set of sense-specific vectors instead of one isolate vector of features; the set of vectors for a word is determined by unsupervised word sense discovery. Thus, ambiguous words can be grouped into homonyms and polysemic words.

A different approach, based on ranking of synsets, has been proposed by other authors (Buitelaar and Sacaleanu, 2001; Kulkarni et al., 2010; Navigli et al., 2011). Navigli et al. (2011) proposed an approach to assign weights to domain synsets from WordNet. The authors first obtained relevant words from texts of the domain, and then such words were used to initialize a random walk over the WordNet graph to get a semantic model for each domain. The semantic models were applied for text categorization and domain-specific WSD tasks. Kulkarni et al. (2010) proposed a knowledge-based approach which retains only domain synsets from WordNet. The WordNet graph was restricted to only those synsets containing words appearing in an untagged corpus. The graph was pruned further to preserve only the largest connected components of the graph. Then, each ambiguous word was disambiguated by using an iterative disambiguation process, only considering those candidate synsets which appear in the *top-k* largest connected elements. Similarly, Buitelaar and Sacaleanu (2001) presented an approach for domain-specific sense assignment, which is, according to our knowledge, the only work related to the proposed approach in this paper. They developed a method for determining domain-specific relevance of GermaNet synsets on the basis of relevance of their words co-occurring within a representative corpus. To seek better results, they added all the direct hyponyms of each synset; in this way each synset was enriched with further lexical information.

Kolte and Bhirud (2008) described an unsupervised approach by using domain information. They used words in the local context to determine the domain of the target word. The approach determines the domain of the target word and the sense corresponding to this domain (from WordNet) is taken as the correct sense. Also Lee and Mit (2011) used domain knowledge; the domain for each word in a sentence was extracted from WordNet by using a domain relevance value; according to the value assigned to each domain, the sense of the ambiguous word is identified.

Other approaches rely on topics from text. Preiss and Stevenson (2013) proposed a topic-based approach by using LDA (Latent Dirichlet Allocation). They created a sense per topic distribution for each LDA topic. The classification of a new document into a topic determines the sense distribution of the words within the topic. Knopp et al. (2013) proposed to use topic models as a way to estimate the distribution of word sense in text, then they used topic-word distributions as a way to derive a semantic representation of ambiguous words in context; these distributions were the input to a k-means algorithm to identify the senses of words. Lau et al. (2014) proposed an unsupervised topic modeling-based approach. They considered each topic as a sense of the target lemma. To extract topics they use a *Hierarchical Dirichlet Process*, and then they computed the similarity between a sense and a topic using the Jensen Shannon divergence between multinomial distributions of the gloss (of ambiguous word) and that of the topic.

Other works are based on semantic graphs (Navigli, 2009b; Navigli and Lapata, 2007; Reddy et al., 2010; Sinha and Mihalcea, 2007) where nodes represent concepts and edges denote the similarities between them. Reddy et al. (2010) presented a graph-based approach, such an approach is based on *Distributional Similarity* (DS) to retrieve neighboring words related to an ambiguous word, then the semantic relatedness between senses of each neighbor word and an ambiguous word is computed (Siddharth and Pedersen, 2003). This information was used to assign weights to nodes in the graph. Finally, the graph was evaluated by using the Personalized PageRank (PPRank) algorithm (Agirre and Soroa, 2009) to disambiguate each instance in the test corpus.

An unsupervised approach was presented by Koeling et al. (2005) to learn predominant senses on different domains. The WSD process was carried out by using a thesaurus from a domain corpus, applying the Lin's method (Lin, 1998) and semantic similarity measures based on WordNet.

The works of McCarthy et al. (2004), Koeling et al. (2005), Reddy et al. (2010), and Guo et al. (2010) are based on the DS technique to construct a distributional thesaurus from a domain-specific auxiliary corpus; for each ambiguous word, its *top-k* nearest neighbors are retrieved. These approaches have the main disadvantage of comparing each sense of an ambiguous word against all the senses of the neighboring words by using semantic relatedness for each pair of senses.

In the literature there exist methods that combine information from a domain-specific auxiliary corpus and a domain-specific test corpus, which seem to be the most suitable for domain-specific WSD (Agirre et al., 2009; Buitelaar and Sacaleanu, 2001), and this is the motivation for the proposed approach.

3. Proposal

For a better understanding of the proposal, some definitions are given next. Moreover, a running example is used along the article to illustrate the proposed approach.

3.1. Definitions

The following definitions will be referenced on the paper.

- A *corpus* is a collection of text documents.
- Let *n* be the total number of ambiguous words in a corpus, an *ambiguous word* is denoted by a_x, $1 \leq x \leq n$. The set of ambiguous words is defined in Equation 1.

$$A = \{a_1, a_2, a_3, \ldots, a_n\} \tag{1}$$

- Let $i_{m_{a_x}}$ be the total number of instances of the ambiguous word a_x, an instance of a_x is denoted by $a_x i_y$, $1 \leq y \leq i_{m_{a_x}}$. The set of instances of the ambiguous word is defined in Equation 2.

$$a_x I = \{a_x i_1, a_x i_2, a_x i_3, \ldots, a_x i_{m_{a_x}}\} \tag{2}$$

The set of all ambiguous words with all its instances in a corpus is defined in Equation 3.

$$AI = \begin{bmatrix} a_1 i_1 & a_1 i_2 & a_1 i_3 & \ldots & a_1 i_{m_{a_1}} \\ a_2 i_1 & a_2 i_2 & a_2 i_3 & \ldots & a_2 i_{m_{a_2}} \\ \ldots & \ldots & \ldots & & \ldots \\ a_n i_1 & a_n i_2 & a_n i_3 & \ldots & a_n i_{m_{a_n}} \end{bmatrix} \tag{3}$$

It is worthy noting that every $a_x i_{m\_a_x}$ could be different, *e.g.* $a_1 i_{m_{a_1}}$ could be different to $a_2 i_{m_{a_2}}$.

- A *synset* is a set of word senses all expressing (approximately) the same meaning (Navigli, 2009a). In this work the resource used for retrieving synsets is WordNet (Miller, 1995). For example, the synsets from WordNet for the verb *reserve* are:
  - (hold back or set aside, especially for future use or contingency) "*they held back their applause in anticipation*"
  - (give or assign a resource to a particular person or cause) "*I will earmark this money for your research*"; "*She sets aside time for meditation every day"*
  - (obtain or arrange (for oneself) in advance) "*We managed to reserve a table at Maxim's*"
  - (arrange for and reserve (something for someone else) in advance) "*reserve me a seat on a flight*"; "*The agent booked tickets to the show for the whole family*"; "*please hold a table at Maxim's*"

Words in brackets correspond to a *gloss*, and words in quotation marks are examples of use.

- A *gloss* is a textual definition of a synset, and can have a set of usage examples (Navigli, 2009a).
- A *test corpus* is a benchmark corpus which is used in the literature for testing WSD approaches. The test corpus has several ambiguous words; each ambiguous word has several instances, which are identified by codes. These instances are previously identified by human experts.
- An *auxiliary corpus* can be seen as a special case of training corpus. It is a corpus created from the *test corpus*, also it contains ambiguous words (from the *test corpus*) but instances are not identified. It gives more contexts for ambiguous words.
- A *domain-specific test corpus* is a corpus for a specific domain (Sports, Biology, Finance, Nutrition, Law, etc.). For easy reading, in the remaining of the paper we refer to *domain-specific test corpus* as *test corpus*.

- A *domain-specific auxiliary corpus* is an auxiliary corpus for a specific domain. For easy reading, in the remaining of the paper we refer to *domain-specific auxiliary corpus* as *auxiliary corpus*.
- Let rf_h a *related feature*, which is a word co-occurring with an ambiguous word a_x in the auxiliary corpus. Let B_{a_x} a *Related Features Vector* for an ambiguous word a_x in the auxiliary corpus defined in Equation 4, where u is the total number of related features of a_x.

$$B_{a_x} = \{rf_1, rf_2, rf_3, \ldots, rf_u\} \tag{4}$$

Let $B'_{a_x i_y}$ the *Instance Related Features Vector* for an instance of an ambiguous word $a_x i_y$ defined in Equation 5, where p is the total number of related features for $a_x i_y$.

$$B'_{a_x i_y} = \{tf_1, tf_2, tf_3, \ldots, tf_p\} \tag{5}$$

Thus, $B'_{a_x i_y} \subseteq B_{a_x}$

- Let tf_h a *test feature*, which is a word co-occurring with an instance of an ambiguous word $a_x i_y$ in the test corpus. Let $C'_{a_x i_y}$ the *Instance Test Features Vector* for an instance of an ambiguous word $a_x i_y$ defined in Equation 6, where r is the total number of test features for $a_x i_y$.

$$C'_{a_x i_y} = \{tf_1, tf_2, tf_3, \ldots, tf_r\} \tag{6}$$

Let C_{a_x} a *Test Features Vector* for an ambiguous word a_x in the test corpus defined in Equation 7, where q is the total number of all test features for a_x.

$$C_{a_x} = \{tf_1, tf_2, tf_3, \ldots, tf_q\} \tag{7}$$

Thus, $C'_{a_x i_y} \subseteq C_{a_x}$

3.2. Running example

For illustrating how the approach works, let us consider the following scenario. We want to disambiguate words from the Sports domain. A common user could select a synsets from WordNet to solve this problem, but it is hard to select the better one because WordNet has many synsets for a word. The proposed approach tries to solve this problem by using two corpora: *test* and *auxiliary*. From these corpora, *features* of ambiguous words are retrieved and used to select the predominant synset for such ambiguous words. Both corpora are used to explain stages of the approach in next subsections.

- *Test corpus*. This is a domain-specific test corpus from the Sports domain (Koeling et al., 2005), all ambiguous words must be disambiguated, one per sentence. The corpus consists of 41 ambiguous words and 3,989 instances of such ambiguous words.
- *Auxiliary corpus*. As described later in Section 3.3.1, an auxiliary corpus is obtained from the Web by using relevant words from the test corpus. Such corpus contains more contexts where ambiguous words (from the test corpus) appear. For example, a partial view of the auxiliary corpus derived from the test corpus of Sports domain is given in Table 1 where three ambiguous words (fan, reserve, and team) appear. The auxiliary corpus will have more contexts (than the test corpus) where ambiguous words occur.

Table 1
Example of sentences from auxiliary corpus; ambiguous words are in bold.

Fans *can* ***reserve*** *a ride by calling (480) 777-9777 or online at www.cleanaircab.com. The more mature athlete can* ***reserve*** *the hardest training for May or June in order to peak later in the season. We stayed committed to playing the* ***team*** *game, and we knew that was what it was going to take to win a gold medal.*

3.3. Approach

The proposed approach works on the lexical sample task of WSD (see Section 1). The general scheme of the proposed approach is depicted in Fig. 1, which consists of four main stages. In the first stage, called *auxiliary corpus generation*, more contexts for ambiguous words are obtained from the Web (sentences where ambiguous words appear). This new set of sentences is named *auxiliary corpus*, which is a special case of training corpus. According to definitions given in Section 3.1, in the second and third stages, called *related features extraction* and *test feature extraction*, features are obtained from the *auxiliary corpus* and *test corpus* respectively. *Related features* are used to predict the predominant sense of the *ambiguous word* and *test features* are used to predict the predominant sense of each *instance of an ambiguous word*. Finally in the fourth stage, called *features integration*, both set of features are integrated to calculate a score for each synset from WordNet for each instance of an ambiguous word (see Section 3.3.4). The score of a synset indicates the predominance of the synset for the instance of the ambiguous word. The synset with more predominance value will be selected as the correct synset for such instance.

3.3.1. Auxiliary corpus generation

The objective in this stage is to obtain more contexts (sentences) for ambiguous words. Relevant words from the *test corpus* are identified by tagging, lemmatizing, removing stopwords, and computing TF · IDF values (Navigli et al., 2011). Only *nouns* and *verbs* with higher TF · IDF values are taken into account because nouns and verbs constitute the core of sentences (Schutz and Buitelaar, 2005).

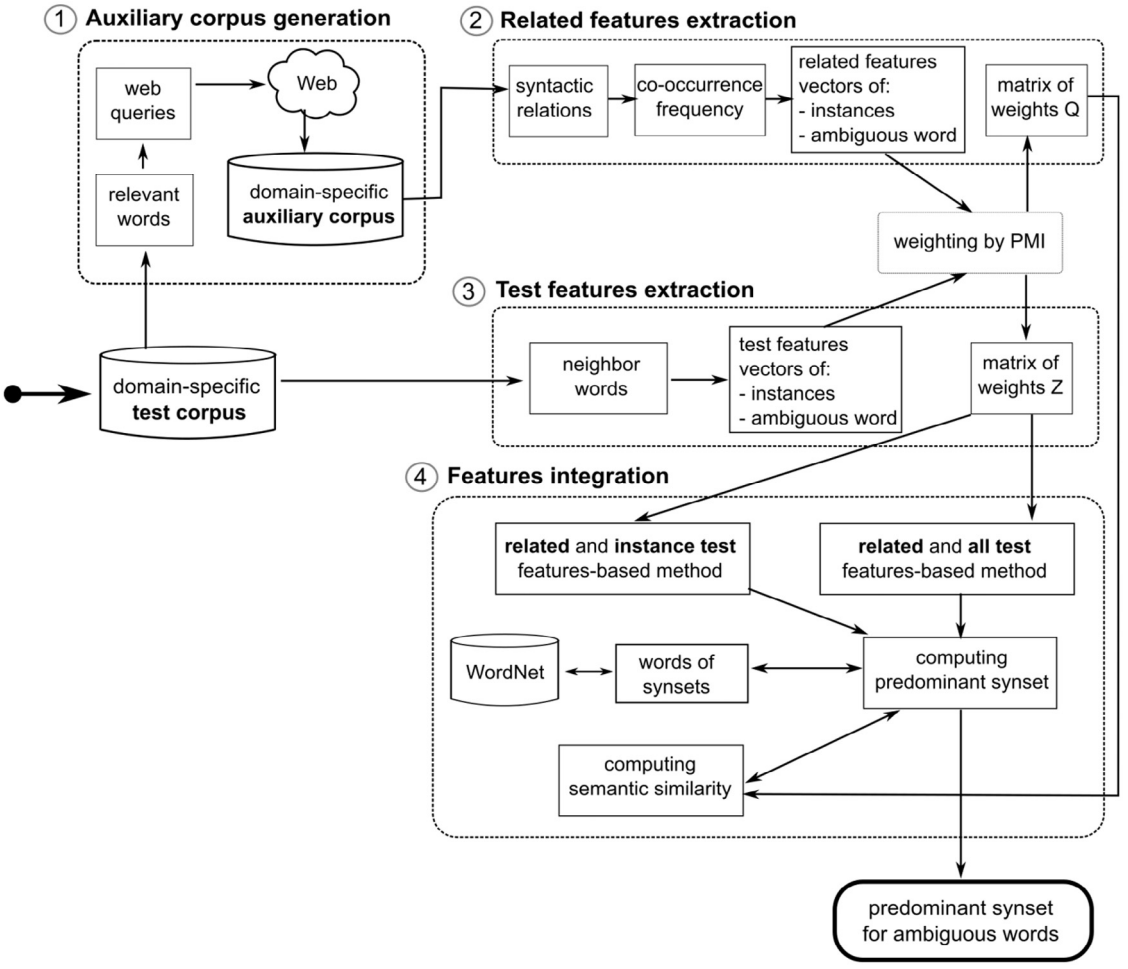

Fig. 1. Approach for domain-specific WSD.

By using relevant words from the test corpus and the domain name, a set of web queries are generated and executed on the Web. The retrieved Web pages are processed to select only domain-relevant Web pages. After a filtering process (eliminating superfluous information: banners, images, ads, etc.), the resulting text of such Web pages constitutes the *auxiliary corpus*.

3.3.2. Related features extraction

The objective in this stage is to obtain *related features* for each ambiguous word from the *auxiliary corpus*. Related features are obtained considering the *Distributional Similarity* approach on the auxiliary corpus. Distributional Similarity considers that words in similar contexts will tend to have the same or related meanings (Harris, 1954; Weeds and Weir, 2005). For example, in the first sentence given in Table 1 (partial view of the auxiliary corpus), the noun *fan* and the verb *reserve* are joined by the syntactic dependency relation *nsubj* (*nominal subject*), as is shown in the first row of Table 2., *i.e.*, both words are related.

To obtain *related features*, the auxiliary corpus is parsed, POS tagged and lemmatized to extract syntactic dependency relations as is shown in Table 2. In this way the syntactic dependency relations allow to map and model domain-specific text to domain-specific knowledge.

The association of pairs of words by syntactic dependency relation is denoted by w_1, *rel*, and w_2, where *rel* denotes a syntactic dependency relation, and w_1 and w_2 represent two syntactically related words. In our approach w_1 can be an ambiguous word (a_x) and w_2 can be a related feature (rf_h), both can be noun or verb. For each word a_x found in the parsed auxiliary corpus the words rf_h*'s* are obtained. Thus, rf_h represents a *related feature* of the word a_x, and *rel* can be some of the following syntactic dependency relations (Marneffe et al., 2006): *direct object* (dobj), *indirect object* (iobj), *noun compound modifier* (nn) or *nominal subject* (nsubj).

After preprocessing the auxiliary corpus, each word has a Part of Speech (POS) category assigned. For example, Table 2 shows the ambiguous words *reserve*, *game*, and *fan*, where the ambiguous word *reserve* (a_x) has two versions according to its POS category: reserve as verb (*reserve_v*) and *reserve* as noun (*reserve_n*). Most of the times a *version* of an ambiguous word matches with an *instance* of the ambiguous word (from the test corpus); if no match occurs such version is ignored in the next tasks. Thus, for practical reasons in the remaining of the paper we refer to a *version of an ambiguous word* in the auxiliary corpus as an *instance of such ambiguous word*.

In the matrix of Table 3, named *matrix of frequencies*, the noun *fan* is a related feature (*a column*) which co-occurs with the instance *reserve_v (a row)*. This matrix represents each instance (*reserve_v, reserve_n, game_n, fan_n, …,* $a_n i_m$) of each ambiguous word and each *related feature* rf_h by rows and columns respectively.

Table 2
Examples of syntactic dependency relations.

Syntactic dependency relation	Frequency
<reserve_v, ***nsubj***, fan_n >	54
<reserve_n, ***nsubj***, athlete_n >	75
<reserve_n, ***nsubj***, game_n >	36
<game_n, ***nn***, team_n >	48
<fan_n, ***nsubj***, team_n >	27
…	…

Bold means to highlight the type of syntactic dependency relation in each line.

Table 3
Example of a matrix of frequencies for instances of ambiguous words co-occurring with related features.

Instance of ambiguous word	team_n	athlete_n	fan_n	…	rf_p
reserve_v	fv_{11}	fv_{12}	fv_{13}	…	fv_{1p}
reserve_n	fv_{21}	fv_{22}	fv_{23}	…	fv_{2p}
game_n	fv_{31}	fv_{32}	fv_{33}	…	fv_{3p}
fan_n	fv_{41}	fv_{42}	fv_{43}	…	fv_{4p}
…	…	…	…	…	…
$a_n i_m$	fv_{m1}	fv_{m2}	fv_{m3}	…	fv_{mp}

Considering the rows of the generated matrix of frequencies, each instance $a_x i_y$ is represented by an *Instance Related Feature Vector* denoted by $B'_{a_x i_y} = \{fv_{y1}, fv_{y2}, fv_{y3}, \ldots, fv_{yp}\}$, where fv_{yh} represents the co-occurrence value when an instance $a_x i_y$ of an ambiguous word a_x appears together with rf_h, where $1 \leq h \leq p$, p is the total number of distinct related features for instance $a_x i_y$. For example the *Instance Related Feature Vector* of instance *reserve_v* in Table 3 is denoted by $B'_{reserve\_v} = \{fv_{11}, fv_{12}, fv_{13}, \ldots, fv_{1p}\}$.

From *Instance Related Feature Vectors* each co-ocurrence value fv_{yh} is used to calculate the degree of association between the instance $a_x i_y$ and the related feature rf_h. This association is computed by applying the *Pointwise Mutual Information* (*PMI*) function (Church and Hanks, 1990), see Equation 8.

$$PMI(a_x i_y, rf_h) = log_2 \frac{P(a_x i_y, rf_h)}{P(a_x i_y, *)P(*, rf_h)} \tag{8}$$

$P(a_x i_y, *)$ represents the sum of occurrences of $a_x i_y$ over all features rf_h, where $1 \leq h \leq p$.

$P(*, rf_h)$ represents the sum of occurrences of rf_h over all instances $a_x i_y$ of the ambiguous word a_x, where $1 \leq y \leq m$, m is the total number of instances of the ambiguous word a_x.

Co-occurrence values are obtained from the matrix of frequencies, where $fv_{a_x i_y, rf_h}$ denotes the co-occurrence frequency of $a_x i_y$ and rf_h, thus $P(a_x i_y, rf_h) = fv_{a_x i_y, rf_h}$.

The *PMI function* compares the number of occurrences between $a_x i_y$ and rf_h with the number of occurrences that $a_x i_y$ and rf_h have independently. Thus, a *matrix of weights* (Q) is generated from the matrix of frequencies of related features. An example of this matrix is given in Table 4. Once the matrix of weights Q is computed, each row in such matrix is named *weighted Instance Related Features Vector*, which corresponds to an instance $a_x i_y$ and is denoted by $wB'_{a_x i_y}$; the weight for a specific related feature rf_h is denoted by $wB'_{a_x i_y}(rf_h)$.

3.3.3. Test features extraction

Following the running example, let us consider the instances of the ambiguous word *reserve* (from the test corpus), as shown in Table 5, which is divided into two parts. The first part shows two instances for the ambiguous word *reserve*, identified by *reserve.n.sports.428* and *reserve.n.sports.434*. After lemmatizing, stopwords are removed from sentences, as shown in the second part of Table 5, then TF · IDF values are calculated for all words in sentences. *Test features* are obtained from the context where each instance of an ambiguous word appears in the test corpus by using a different strategy than the one used for the related features extraction process.

Table 4
Weighted related features representation (matrix Q).

Instance of ambiguous word	team_n	athlete_n	fan_n	…	rf_p
reserve_v	0.00	2.56	0.00	…	wfv_{1p}
reserve_n	0.00	0.00	0.89	…	wfv_{2p}
game_n	2.81	0.00	0.00	…	wfv_{3p}
fan_n	1.79	0.00	0.00	…	wfv_{4p}
…	…	…	…	…	…
$a_n i_m$	wt_{m1}	wt_{m2}	wt_{m3}	…	wfv_{mp}

Table 5
Example of instances for the ambiguous word *reserve* in the test corpus.

ID instance	Context
reserve.n.sports.428	*Athletic Bilbao beat Madrid FC 5-0 in 1915 while in 1980 Real Madrid thrashed Castilla; their* ***reserve*** *team now known as Real Madrid b, 6-1.*
reserve.n.sports.434	*Hardy was expected to take his place on Australia is* ***reserves*** *bench at Ballymore after first-choice test full-back Matt Burke withdrew on Tuesday because of a groin injury.*
…	…
reserve.n.sports.428	*athletic beat madrid real madrid thrash castilla* ***reserve*** *team now know real madrid*
reserve.n.sports.434	*hardy be expect take place australia be* ***reserve*** *bench first choice test full back matt burke withdraw tuesday groin injury*
…	…

Table 6
Weighted test feature representation for the ambiguous word *reserve* (matrix Z).

Instance of ambiguous word	thrash	castilla	team	now	australia	be	bench	first
reserve.n.sports.428	8.7	6.7	11.4	10.9	0.0	0.0	0.0	0.0
reserve.n.sports.434	0.0	0.0	0.0	0.0	6.8	5.4	10.2	7.4

To obtain *test features*, instances of ambiguous word are identified on sentences from the test corpus. The *context* of an instance of ambiguous word is defined by using a *context window* on the sentence where the ambiguous word appears. On each sentence a *context window* denoted by *CW* is defined for the instance $a_x i_y$, the size of the context window is $2\beta + 1$ words (Islam and Inkpen, 2006). Each word in *CW* is named a *test feature* (tf_h) and is represented by using the *Bag of Words* model. For example, given a sentence denoted by $sentence_i = \{\ldots, tf_{-3}, tf_{-2}, tf_{-1}, \boldsymbol{a_x i_y}, tf_1, tf_2, tf_3, \ldots\}$ and $\beta = 2$, the set of words (*test features*) in the context for an instance $a_x i_y$ consists of $\{tf_{-2}, tf_{-1}, tf_1, tf_2\}$ which could include nouns, verbs, adjectives, or adverbs; note that $a_x i_y$ is not included in the context window. Thus, for each instance $a_x i_y$, a list of *test features* tf_h contained in its context is retrieved from sentences of the test corpus, h is the total number of test features for $a_x i_y$.

Each instance $a_x i_y$ is represented by its context as an *Instance Test Features Vector* denoted by $C'_{a_x i_y} = \{fv_{y1}, fv_{y2}, fv_{y3}, \ldots, fv_{yr}\}$; each test feature tf_h is associated with an instance $a_x i_y$ by its frequency fv_{yh}. For example, according to Table 5 (second part), the *Instance Test Features Vector* for instance *reserve.n.sports.428* is represented by $C'_{reserve.n.sports.428} = \{fv_1, fv_2, fv_3, fv_4\}$, where fv_1, fv_2, fv_3, and fv_4 correspond to frequencies of test features *thrash*, *Castilla*, *team*, and *now* respectively.

The union of all *test features* of every instance of an ambiguous word a_x constitutes a *Test Features Vector*, which is denoted by $C_{a_x} = \bigcup_{y=1}^{m} C'_{a_x i_y}$, m is the total number of instances of the ambiguous word a_x. For example, according to Table 5 (second part), the *Test Features Vector* for the ambiguous word *reserve* is represented as $C_{reserve} = \{fv_1, fv_2, fv_3, fv_4, fv_5, fv_6, fv_7, fv_8\}$, where fv_1, fv_2, fv_3, fv_4, fv_5, fv_6, fv_7, and fv_8 correspond to frequencies of test features: *thrash*, *castilla*, *team*, *now*, *australia*, *be*, *bench*, and *first* respectively.

For representing the relevance of each test feature tf_h respect to an ambiguous word a_x, the association degree between the ambiguous word a_x and a test feature tf_h is also computed by using the *PMI* function (Equation 8). In this case $P(a_x, tf_h)$ is the frequency between the ambiguous word a_x and the test feature tf_h. $P(a_x, *)$ and $P(*, tf_h)$ are the sums of occurrences in the whole test corpus of a_x and tf_h respectively, i.e. all words in the lemmatized sentence where the ambiguous word a_x appears are taken into account (as illustrated in the second part of Table 5). The association values for the ambiguous word a_x are represented as *weighted Test Features Vector* denoted by wC_{a_x}; association values for a specific $a_x i_y$ is represented by *weighted Instance Test Features Vector* denoted by $wC'_{a_x i_y}$, a weight for a specific test feature tf_h of an instance $a_x i_y$ is denoted by $wC'_{a_x i_y}(tf_h)$. As a result of applying this process, a *matrix of weights (Z)* is obtained, as Table 6 shows for the ambiguous word *reserve*.

3.3.4. Features integration

After obtaining associated words to the ambiguous word (*related features*) and associated words to instances of the ambiguous word (*instance test features*), both sets of features are integrated to select the predominant sense (synset from WordNet) for instances of ambiguous words. This task is achieved by two methods. The first method uses *related features* of the ambiguous word and *test features* of an instance of the ambiguous word (*instance test features*). The second method uses *related features* of the ambiguous word and *all test features* of the ambiguous word (considering all instances of such ambiguous word). Considering an instance of an ambiguous word, both methods calculate a score for each synset from WordNet of such instance. Both methods determine the association degree of words of each synset and the instance of the ambiguous word. The method also computes the semantic similarity between such words and the instance. If the words of a synset correspond to *related* or *test* feature of the instance then the score of such synset will be increased.

After obtaining *related* and *test* features from the auxiliary and test corpus respectively, the *matrices of weights* were generated (see Sections 3.3.2 and 3.3.3). The *matrix of weights Q* (from related features) is used to measure the semantic similarity between the instance of the ambiguous word and each word from synsets. The *matrix of weights Z* (from test features) is used to seek the association degree between an instance of the ambiguous word and each

word from synsets. If both words share a large number of features, the semantic similarity between both words will be larger (Budanitsky and Hirst, 2006).

3.3.4.1. Semantic similarity between words. Semantic similarity is a reliance value that reflects the semantic relation between words. In the literature there exist several methods for determining semantic similarity between words: semantic network-based and corpus-based statistical approaches (Jiang and Conrath, 1997), Web-based (Bollegala et al., 2007), WordNet-based (Li et al., 2003; Patwardhan and Pedersen, 2006; Richardson et al., 1994), taxonomy-based (Resnik, 1995) and more recently ontology-based (Moreno et al., 2013; Sánchez et al., 2012; Selvi and Gopalan, 2007; Serra et al., 2014; Wróblewska et al., 2013).

The proposed approach uses an unsupervised corpus-based technique relying on statistical associations to calculate semantic similarity between words. Such information is extracted from the *matrix of weights Q* (see Table 4) of related features because the auxiliary corpus has more contexts for ambiguous words than the test corpus. This is achieved by means of the *cosine of Pointwise Mutual Information* similarity function (Equation 9) (Terra and Clarke, 2003; Zhao and Lin, 2005), where $a_x i_1$ and $a_x i_2$ are instances of ambiguous words (rows) from the *matrix of weights Q*, but can be any pair of words; values closer to one indicate more similarity whereas values close to zero represent less similarity.

$$sim_{cosPMI}(a_x i_1, a_x i_2) = \frac{\sum_{rf_j \in Q[a_x i_1] \cap Q[a_x i_2]} PMI(a_x i_1, rf_j)\, PMI(a_x i_2, rf_j)}{\sqrt{\sum_{rf_j \in Q[a_x i_1] \cap Q[a_x i_2]} PMI(a_x i_1, rf_j)^2}\sqrt{\sum_{rf_j \in Q[a_x i_1] \cap Q[a_x i_2]} PMI(a_x i_2, rf_j)^2}} \tag{9}$$

3.3.4.2. Related and instance test features-based method. This method is named WSD_{RinsTF}. Given that wC_{a_x} contains all *weighted test features* for the ambiguous word a_x, then $wC'_{a_x i_y}$ (for an instance $a_x i_y$) is a subset of wC_{a_x}. Therefore, the weight for each test feature tf_h that measures the association degree with a particular instance $a_x i_y$ is retrieved from wC_{a_x} by $wC'_{a_x i_y}(tf_h)$. For example, in Table 7 we can see the weights w_h of each test feature tf_h (*thrash*, *Castilla*, *team*, *now*) for the instance *reserve.n.sports.428*, which were retrieved from Table 6.

For obtaining the most relevant synset for an instance $a_x i_y$, the following steps are executed: (1) all synsets for $a_x i_y$ are retrieved from WordNet, which is denoted by $S = \{s_1, s_2, \ldots, s_c\}$, where c is the total number of synsets for $a_x i_y$, and (2) words of glosses of each synset (which can be nouns, verbs, adjectives, or adverbs) are parsed, tagged according to its POS category, and stopwords are removed. Formally, let $s_k = \{t_1, t_2, \ldots, t_d\}$ be the set of words in the gloss of synset k, $1 \le k \le c$, d is the total number of words in the gloss, and let $wC'_{a_x i_y} = \{wfv_1, wfv_2, \ldots, wfv_r\}$ be the set of weighted values for test features for the instance of ambiguous word ($a_x i_y$) where r is the total number of test features for $a_x i_y$; then the predominance of each synset s_k for instance $a_x i_y$ is calculated by using Equation 10.

$$W(s_k, a_x i_y) = \sum_{j=1}^{|s_k|} wC'_{a_x i_y}(t_j) + sim_{cosPMI}(a_x i_y, t_j) \tag{10}$$

Related and test features of $a_x i_y$ are used to identify words from the gloss of s_k. Each word t_j in the gloss of s_k is searched in $wC'_{a_x i_y}$; if $t_j \in wC'_{a_x i_y}$ then t_j is an *instance test feature* of the instance $a_x i_y$ and its weight is retrieved by $wC'_{a_x i_y}(t_j)$, which indicates the association between the instance $a_x i_y$ and the word t_j. Given that $t_j \in wC'_{a_x i_y}$, a semantic similarity value between $a_x i_y$ and t_j is computed by $sim_{cosPMI}(a_x i_y, t_j)$. Finally the association degree between the instance $a_x i_y$ and the word t_j is added to the semantic similarity of the instance $a_x i_y$ and the word t_j to obtain the final score of synset s_k.

Table 8 shows synsets for the ambiguous word *reserve* (instance *reserve.n.sports.428*). The first column denotes the number of synset, the second column shows the key of synset in WordNet, and the third column corresponds to

Table 7
Weighted test features representation for instance *reserve.n.sports.428*.

Instance of ambiguous word	thrash	castilla	team	now
reserve.n.sports.428	8.7	6.7	11.4	10.9

Table 8
Synsets for the ambiguous word reserve (instance *reserve.n.sports.428*).

No.	Key	Contextual definition of gloss
$synset_1$	4900121	*formality_n property_n manner_n*
$synset_2$	13368052	*kept_v back_r save_v future_a use_n special_a purpose_n*
$synset_3$	10671042	*athlete_n play_v only_r starter_a* ***team_n*** *be_v replace_v*
$synset_4$	13759773	*medicine_n potential_a capacity_n respond_v order_n maintain_v vital_a function_n*
$synset_5$	8587174	*district_n be_v reserve_v particular_a purpose_n*
$synset_6$	8206460	*armed_a force_n be_v active_a duty_n can_v call_v emergency_n*
$synset_7$	4652438	*trait_n be_v uncommunicative_a volunteer_v more_a necessary_a*

Bold means to highlight that, after an extraction process, only line 3 (synset_3) contain the word 'team_n' which is has a relation to word 'reserve' (from Table 7).

words of the gloss after preprocessing. For the running example, the word *team* of $synset_3$ is found in the context of *reserve*, therefore the score of $synset_3$ is increased. Then, for each $t_j \in s_k$, a semantic similarity value is computed between the word t_j and the instance $a_x i_y$. For example, in the case of *reserve_n* with *team_n*, a semantic similarity value is computed and associated to $synset_3$. Thus, after this process is completed, the synset with maximum score is returned as the predominant synset for the instance *reserve.n.sports.428*.

3.3.4.3. Related and all test features-based method. This method is named WSD_{RallTF}. In this case, instead of disambiguate each instance $a_x i_y$ of the ambiguous word a_x by using each of its sentences in which it occurs (contexts), this method uses all *test features* of the ambiguous word a_x (all contexts of a_x), as shown in Table 6 for the ambiguous word *reserve*. In this way, we use a larger contextual representation for each ambiguous word a_x to find its predominant sense.

Similar to the WSD_{RinsTF} method, the same steps are executed, words from glosses of synsets of a_x are retrieved, and Equation 10 is used, but using all *weighted test features* for a_x (wC_{a_x}). In this case the first term of Equation 10 represents $wC_{a_x}(t_j)$ instead of $wC'_{a_x i_y}(t_j)$. If $t_j \in wC_{a_x}$ then t_j is a *test feature* of the ambiguous word a_x and its association value is retrieved by using $wC_{a_x}(t_j)$. The semantic similarity is now computed by $sim_{cosPMI}(a_x, t_j)$. The degree of association between a_x and t_j is added to the semantic similarity of a_x and t_j to obtain the final score of synset s_k.

The Algorithm 1 in Appendix A illustrates both methods to disambiguate instances of an ambiguous word.

4. Results

In this section the employed resources, evaluation scenario, results, and remarks of the implementation of the method are described to show the performance of the proposed approach.

4.1. Test corpora

For testing purposes, the proposed method was tested on several specific-domain corpora such as Nutrition, Law, Sports, and Finance. Also was tested on one general-domain corpus, the British National Corpus (BNC).

For comparison purposes the method was tested on Sports, Finance, and BNC corpora, which are commonly used as benchmarks in the state-of-the-art for domain-specific WSD. The experiments were carried out on the gold standard dataset employed by Koeling et al. (2005). This dataset comprises three corpora from Sports and Finance domains, and a general-domain corpus, the BNC. Each ambiguous word from the dataset has about 100 occurrences. Table 9 summarizes general information of test corpora. The total number of occurrences of ambiguous words for the whole dataset is 12,105. The inter-tagger agreement refers to the proportion of human experts who agreed about assigning a particular sense to an ambiguous word. Since each ambiguous word has several senses the 100% of agreement is never achieved. These corpora have been used in previous works in the state of the art (Agirre and Lopez, 2009; Agirre et al., 2009; Koeling et al., 2005; Lau et al., 2014; Navigli et al., 2011).

Table 9
General information about the used corpora.

Corpus	Ambiguous words	Instances of ambiguous words	Average polysemy	Inter-tagger agreement
Sports	41	3989	6.7	65%
Finance	41	4021	6.7	69%
BNC	41	4095	6.7	60%

Table 10
Examples of queries generated to obtain a domain-specific auxiliary corpus.

Domain name	Query
Finance	bank and market + *finance*
Sports	olympic and stadium + *sports*
BNC	powerpc and file

4.2. Experimental setting

This section describes how *related* and *test features* were obtained from an *auxiliary corpus* and a *test corpus* respectively.

4.2.1. Configuration for related features

In the proposed approach the availability of an *auxiliary corpus* for the domain of each test corpus is a very important point. To obtain related features it is necessary to get a representative auxiliary corpus from the domain. The representative auxiliary corpus is retrieved from the Web by executing a set of web queries on a search engine by using relevant words extracted from the *test corpus*. The extraction of relevant words was achieved by using Stanford parser[2] and Stanford POS tagger[3] tools (Toutanova et al., 2003). Relevant words (nouns and verbs) with TF · IDF values > 0.65 were used as seed to generate Web queries of length two according to Iosif and Potamianos (2010). Query formulation is a crucial point to retrieve relevant information from the Web; thus to make more precise queries the domain name was added to each query (see Table 10). It is important to note that BNC is a general-domain corpus; therefore it does not use a domain name.

The Google search engine was employed retrieving the top 25 documents from the Web per each query, including HTML and PDF files. These files were converted to plain text by using the Apache Tika tool (Tika, 2014) and split by sentences by using the Apache OpenNLP tool (OpenNLP, 2014). The name of the domain of the test corpus was added to each Web query to retrieve Web pages from the same domain; additionally the semantic similarity of each webpage respect to the test corpus was computed. For this, from the text of each webpage or HTML/PDF file were identified nouns and verbs by using Stanford parser and Stanford POS tagger tools. Nouns and verbs of each webpage or HTML/PDF file were compared to relevant nouns and verbs of the test corpus by using the DISCO[4] (extracting DIStributionally related words using CO-occurrences) tool. DISCO computes semantic similarity between pairs of words by using information from Wikipedia. Only Web pages or HTML/PDF files with semantic similarity > 0.65 were retained, its text was added to the auxiliary corpus. The content of the auxiliary corpus was verified to maintain valid sentences (sentences containing nouns related to verbs). The corpus was parsed, POS tagged, and lemmatized by using the Stanford parser and Stanford POS tagger tools to extract syntactic dependency relations of the whole text; these relations were used to identify related features. It is important to point out that this process is performed without any human intervention.

[2] http://nlp.stanford.edu/software/lex-parser.shtml (visited on February 2016).
[3] http://nlp.stanford.edu/software/tagger.shtml (visited on February 2016).
[4] http://www.linguatools.de/disco/disco_en.html (visited on February 2016).

4.2.2. Configuration for test features

Words within the neighborhood of an instance of an ambiguous word have an important role in the process of disambiguation, because they contribute strongly in the ranking of synsets to determine the predominant sense of each instance of an ambiguous word. To extract *test features* from the context of an instance of an ambiguous word, the test corpus was tagged by using Stanford POS tagger to assign a grammatical category to each word in sentences. To select *test features* a context window size was defined as explained in Section 3.3.3, $2\beta + 1$, with $\beta = 5$, i.e. 5 words on the right and 5 words on the left of each instance of the ambiguous word were retrieved from sentences where the instance appears.

4.3. Analysis of results

For comparing results, *Precision* (*P*) and *Recall* (*R*) were used as evaluation measures. Tables 11 and 12 show summaries of results of partial tasks on test and auxiliary corpora respectively.

Table 13 shows the final results of the proposed approach on test corpora for comparison purposes. These results are separated in three sections. The first section shows the baseline, which is a heuristic-based approach on taking as correct the first synset found in WordNet for the ambiguous word. The second section shows the results by using the *related and instance test features-based method* (WSD_{RinsTF}). The third section shows the results by using the *related and all features-based method* (WSD_{RallTF}). In the last two sections better values are in bold. The results were compared against those reported in the literature (Agirre et al., 2009; Koeling et al., 2005; Lau et al., 2014; Navigli et al., 2011). Results of the proposed approach represent an improvement over the results presented by Lau et al. (2014)

Table 11
Results of partial tasks on test corpus.

Corpus	Sentences	TVN	TRVN	Queries	TRWF
Sports	3989	7774	1876	1,884,711	33,794
Finance	4021	5466	1082	930,930	28,897
BNC	4095	6042	1976	1,945,378	35,985

TVN, total of verbs and nouns from test corpus (with repetitions); TRVN, total of relevant nouns and verbs from test corpus (with repetitions); TRWF, total of retrieved Web pages and HTML/PDF files.

Table 12
Results of partial tasks on auxiliary corpus.

Corpus	Sentences	TVN	Syntactic dependency relations
Sports	162,278	386,468	632,214
Finance	274,557	673,682	1,479,899
BNC	364,293	942,880	1,445,432

TVN, total of verbs and nouns from auxiliary corpus (with repetitions).

Table 13
Results of experiments by integrating auxiliary and test corpora.

Method	Sports		Finance		BNC	
	P	R	P	R	P	R
WordNet first-sense baseline	24.1	24.1	38.5	38.5	40.4	40.4
$WSD_{RinstTF}$	**61.9**	**54.8**	**63.8**	**63.2**	30.8	27.9
Topic-based WSD (Lau et al., 2014)	42.2	–	55.5	–	37.6	–
Semantic model vector (Navigli et al., 2011)	–	52.7	–	58.2	–	–
Predominant sense (Koeling et al., 2005)	49.7	–	43.7	–	**40.7**	–
WSD_{RallTF}	**66.4**	**65.7**	**67.2**	**66.4**	31.6	31.0
Personalized PageRank (Agirre et al., 2009)	–	51.5	–	59.3	–	**40.7**

Bold means to highlight the higher values on the comparative.

Table 14
Results of experiments by using auxiliary and test corpus in isolate manner.

Isolate used corpus	Sports		Finance		BNC	
	P	R	P	R	P	R
Test corpus	53.0	8.1	50.1	8.5	**49.0**	8.6
Auxiliary corpus	**63.2**	**55.0**	**65.7**	**64.9**	26.7	**24.1**

Bold means to highlight the higher values on the comparative.

who used topic models, Navigli et al. (2011) who used the Semantic Model Vector, Agirre et al. (2009) who used the Personalized Page Rank algorithm, and the results given by Koeling et al. (2005) who used predominant sense acquisition from a domain-specific corpus. The obtained results by the proposed approach on the Sports and Finance domains outperform those reported in the literature. Nevertheless, the obtained results on the general-domain corpus (BNC) presented a low precision, which is below than the baseline (see the Section 4.4 for more comments).

Results of the evaluation using independently the test and auxiliary corpora (without integrating them) are shown in Table 14. We observed that the contribution of the isolate test corpus is poor in terms of Recall, and better results were obtained when the isolate auxiliary corpus is used.

Comparing the combined use of corpora (Table 13) and the isolate use of corpora (Table 14), we can see that the WSD_{RinsTF} method obtained lower results than using the isolate auxiliary corpus, but better results were obtained with the WSD_{RallTF} method by using the combination of corpora. WSD_{RallTF} has better results than WSD_{RinsTF} because WSD_{RallTF} uses all contexts of all instances of ambiguous words and semantic information from the Web, while WSD_{RinsTF} uses only context of a particular instance of the ambiguous word and semantic information from the Web.

4.4. Remarks

It is important to emphasize the following:

- Applying the WSD_{RinsTF} method. The results of this method are close for the Sports and Finance domains, but it has poor performance on the general-domain corpus BNC (as expected). It is important to highlight that Web queries generated for BNC do not include the domain name (see Table 10), and then results of Web searches were not limited to some specific domain, thus the text is dispersed. In this manner, few sentences contain variants of ambiguous words. To obtain better results the general-domain corpus should be larger and more diverse to cover many topics to provide more sentences where ambiguous words appear. With more sentences containing variants of ambiguous words the predominance of its senses will be greater.
- Applying the WSD_{RallTF} method. The results of this method confirm the affirmation given by Agirre et al. (2009): *using related words instead of the actual occurrence contexts yields better results on the domain dataset*. Thus, the approach based on the *all test features* got better results on the Sports and Finance domains. This happened because the number of words in the *Instance Test Feature Vector* is limited by a context window considering 10 words in the neighborhood of an ambiguous word, whereas the *Test Feature Vector* represents all contexts of all instances of an ambiguous word and consequently larger contexts. There is more probability that a word is contained in a synset of an ambiguous word ($t_j \in s_k$) by using the *Test Feature Vector*, which is not the case by using the *Instance Test Feature Vector*. This fact contributes to a higher score for synsets considering the *Test Feature Vector* in the WSD_{RallTF} method.

5. Discussion

The proposed approach relies on the knowledge extracted from the auxiliary and test corpora. From the test corpus lexical information is obtained (neighbors of ambiguous words – *test features*), and from the auxiliary corpus

semantic information is obtained (by using dependency relations – *related features*). It is important to note that the *matrix of weights of related features Q* (as represented in Table 4) is used to predict the predominant sense of an ambiguous word by using the auxiliary corpus. This information is combined with the *matrix of weights of test features Z* (as represented in Table 7), which predicts the sense for each instance of an ambiguous word by using the test corpus. Both matrices have a crucial role in our proposal because intersections of rows and columns of both matrices denote the degree of association between an ambiguous word and a related feature, for *matrix of weights Q*; and an instance of an ambiguous word and a test feature, for *matrix of weights Z*. Since the auxiliary corpus is larger than the test corpus, the auxiliary corpus has more contexts for ambiguous words, which is useful for extracting dependency relations (semantic information), thus the *matrix of weights Q* is used to compute semantic similarity by the *cosine of Pointwise Mutual Information* similarity function (Equation 9). The method gives better results when the auxiliary corpus is larger; the semantic information is increased. We think that the module *auxiliary corpus generation* could be enhanced by polishing the approach for filtering text from the Web. It is desirable to retrieve more contexts but not to increase the number of domains.

The proposed approach was inspired by the affirmation of Gale *et al.* (1992) – *WSD is very dependent on the domain of application* – and by the approach proposed by Agirre et al. (2009), who demonstrated that using related words as context yields better results on domain-specific corpora. This motivated us to propose the integration of lexical information (from test corpus) and semantic information (from the auxiliary corpus). This approach performed well on Sports and Finance domains. The experiments showed that by using *all test features* the Precision is enhanced (see Table 13).

6. Conclusions

This paper presents an approach for domain-specific WSD for the lexical sample task. The basic idea is that words co-occurring with an ambiguous word in the test corpus are profitably to identify words from contextual definitions of such ambiguous word. From the *test corpus*, with ambiguous words, relevant words are identified. Based on such relevant words an *auxiliary corpus* from the Web is obtained; from this corpus *related features* are retrieved. From the *test corpus* neighbor words of instances of ambiguous words are obtained, and these words are named *test features*. For each set of *features* a matrix of weights is computed. From these matrices, the matrix of weights (Q) of related features is used to compute the semantic similarity between the ambiguous word and a feature, or between the instance of an ambiguous word and a feature, where feature can be *related* or *test*. These semantic similarity values are added to its association degree to disambiguate each instance of the ambiguous word in the test corpus. The results of applying the approach demonstrate that in integrating such information it is possible to construct a competitive method for domain-specific WSD according to results reported in the literature.

Although the proposed approach has been tested and evaluated on the Sports, Finance, and BNC corpora obtaining comparable results, more experiments on other domains are necessary to ensure the aplicability of the approach on distinct domains. Partial experiments on Nutrition and Law domains were also carried out obtaining good results. According to this experimentation, the developed prototype showed a reliable and reusable performance. The proposed approach does not require adaptations to process unknown corpora; the only requirements are that the test corpus must be domain-specific and documents must be in the English language. The developed prototype has as drawback that by using the Web as linguistic resource, sometimes the generated auxiliary corpus could be poor; this process requires longer time for searching relevant Web pages.

Appendix A

The general algorithm of the specific-domain WSD approach is given below. The algorithm receives as input the following parameters:

- wC_{a_x}, the matrix of weights of *all test features* of a_x
- $wC'_{a_x i_y}$, the matrix of weights of *instance test features* of $a_x i_y$
- $a_x i_y$, an instance of the ambiguous word a_x

```
Requires: (wC$a_x$), (wC'$a_x i_y$), ($a_x i_y$)
Output: The predominant synset for instance $a_x i_y$
{Obtain synsets of $a_x i_y$ from WordNet}
S = TotalSynsets($a_x i_y$)
 for all $s_k$ ∈ S do
   scoreSense($s_k$) = 0
   $wgls_k$ = getWordsFromGloss($s_k$)
   for all $t_j$ ∈ $wgls_k$ do
     if(disambiguating by using instance test features) then
        if $t_j$ ∈ wC'$a_x i_y$ then
           scoreSense($s_k$) += wC'$a_x i_y$($t_j$)
           {obtain semantic similarity for words}
           scoreSense($s_k$) += $sim_{cosPMI}$($a_x i_y$,$t_j$)
       end if
     else
       {disambiguating by using all test features}
       if $t_j$ ∈ wC$a_x$ then
           scoreSense($s_k$)  += wC$a_x$($t_j$)
           {obtain semantic similarity for words}
           scoreSense($s_k$) += $sim_{cosPMI}$($a_x$,$t_j$)
       end if
     end if
   end for
 end for
return argmax from scoreSense
```

References

Agirre, E., Edmonds, P., 2007. Word Sense Disambiguation: Algorithms and Applications. Springer, Berlin.

Agirre, E., Lopez, O., 2008. On robustness and domain adaptation using SVD for word sense disambiguation. In: International Conference on Computational Linguistics. pp. 17–24.

Agirre, E., Lopez, O., 2009. Supervised domain adaption for WSD. In: Conference of the European Chapter of the Association for Computational Linguistics. pp. 42–50.

Agirre, E., Soroa, A., 2009. Personalizing PageRank for word sense disambiguation. In: Conference of the European Chapter of the Association for Computational Linguistics. pp. 33–41.

Agirre, E., Lopez, O., Soroa, A., 2009. Knowledge-based WSD on specific domains: performing better than generic supervised WSD. In: Proceedings of the 21st International Joint Conference on Artifical Intelligence. pp. 1501–1506.

Agirre, E., De Lacalle, O.L., Fellbaum, C., Marchetti, A., Toral, A., Vossen, P., et al., 2010. SemEval-2010 task 17: all-words word sense disambiguation on a specific domain. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 75–80.

Bollegala, D., Matsuo, Y., Ishizuka, M., 2007. Measuring semantic similarity between words using web search engines. In: Proc. 16th international conference on World Wide Web. Alberta, Canada, pp. 757–766.

Budanitsky, A., Hirst, G., 2006. Evaluating WordNet-based measures of lexical semantic relatedness. Comput. Ling. 32 (1), 13–47.

Buitelaar, P., Sacaleanu, B., 2001. Ranking and selecting synsets by domain relevance. In: Proceedings of the North American Chapter of the Association for Computational Linguistics Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations.

Church, K.W., Hanks, P., 1990. Word association norms, mutual information, and lexicography. Comput. Ling. 16 (1), 22–29.

Escudero, G., Márquez, L., Rigau, G., 2000. An empirical study of the domain dependence of supervised word sense disambiguation systems. In: Proceedings of the joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, *Association for Computational Linguistics*, vol. 13. pp. 172–180.

Faralli, S., Navigli, R., 2012. A new minimally-supervised framework for domain word sense disambiguation. In: Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1411–1422.
Gale, W.A., Church, K.W., Yarowsky, D., 1992. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In: Association for Computational Linguistics. pp. 249–256.
Guo, Y., Che, W., He, W., Liu, T., 2010. HIT-CIR: an unsupervised WSD system based on domain most frequent sense estimation. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 407–410.
Harris, Z., 1954. Distributional structure. In: The Philosophy of Linguistics, vol. 10. no. 23. pp. 146–162.
Iosif, E., Potamianos, A., 2010. Unsupervised semantic similarity computation between terms using web documents. In: IEEE Transactions on Knowledge Data Engineering, vol. 22. no. 11. pp. 1637–1647.
Islam, A., Inkpen, D., 2006. Second order co-occurrence PMI for determining the semantic similarity of words. In: Proceedings of the International Conference on Language Resources and Evaluation. pp. 1033–1038.
Jiang, J.J., Conrath, D.W., 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In: Proc. of the Int'l. Conf. on Research in Computational Linguistics. pp. 19–33.
Knopp, J., Volker, J., Ponzetto, S.P., 2013. Topic modeling for word sense induction. In: *Proc.*25th International Conference on Language Processing and Knowledge in the Web. Lecture Notes in Computer Science, vol. 8105. pp. 97–103.
Koeling, R., McCarthy, D., Carroll, J., 2005. Domain-specific sense distributions and predominant sense acquisition. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. pp. 419–426.
Kolte, S.G., Bhirud, S.G., 2008. Word Sense Disambiguation Using WordNet Domains. In: *Proc.* First International Conference on Digital Object Identifier. IEEE, pp. 1187–1191.
Kulkarni, A., Khapra, M.M., Saurabh, S., Bhattacharyya, P., 2010. CFILT: resource conscious approaches for all-words domain specific WSD. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 421–426.
Lau, J.H., Cook, P., McCarthy, D., Gella, S., Baldwin, T., 2014. Learning word sense distributions, detecting unattested senses and identifying novel senses using topic models. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 1. pp. 259–270.
Lee, W.J., Mit, E., 2011. Word sense disambiguation by using domain knowledge. In: *Proc.* IEEE International Conference on Semantic Technology and Information Retrieval. pp. 28–29.
Li, Y., Bandar, Z.A., Mclean, D., 2003. An approach for measuring semantic similarity between words using multiple information sources. IEEE Trans. Knowl. Data Eng. 15 (4), 871–882.
Lin, D., 1998. Automatic retrieval and clustering of similar words. In: Proceedings of the 17th International Conference on Computational Linguistics, vol. 2. pp. 768–774.
Marneffe, M.C., McCartney, B., Manning, C.D., 2006. Generating typed dependency parses from phrase structure trees. In: Proceedings International Conference on Language Resources and Evaluation. pp. 449–454.
Martínez, D., Agirre, E., 2000. One sense per collocation and genre/topic variations. In: Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, vol. 13. pp. 207–215.
McCarthy, D., 2009. Word sense disambiguation: an overview. In: Language and Linguistics Compass, vol. 3, no. 2. p. 537.
McCarthy, D., Koeling, R., Weeds, J., 2004. Finding predominant word senses in untagged text. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. pp. 279–286.
McCarthy, D., Koeling, R., Weeds, J., 2007. Unsupervised acquisition of predominant word senses. Comput. Ling. 33 (4), 553–590.
Mihalcea, R., Chklovski, T., Kilgarriff, A., 2004. The Senseval-3 English lexical sample task. In: Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text. Barcelona, Spain, pp. 25–28.
Miller, G.A., 1995. WordNet: A Lexical Database for English. Communications of the ACM, vol. 38, No. 11. pp. 39–41.
Moreno, A., Isern, D., López Fuentes, A.C., 2013. Ontology-based information extraction of regulatory networks from scientific articles with case studies for Escherichia coli. Expert Syst. Appl. 40 (8), 3266–3281.
Navigli, R., 2009a. Word sense disambguation: a survey. ACM Comput. Surv. 41 (2), 10:1–10:69.
Navigli, R., 2009b. Using cycles and quasi-cycles to disambiguate dictionary glosses. In: Conference of the European Chapter of the Association for Computational Linguistics. pp. 594–602.
Navigli, R., Lapata, M., 2007. Graph connectivity measures for unsupervised word sense disambiguation. In: Proceedings of the International Joint Conference on Artificial Intelligence. Hyderabad, India, pp. 1683–1688.
Navigli, R., Faralli, S., Soroa, A., de Lacalle, O., Agirre, E., 2011. Two birds with one stone: learning semantic models for text categorization and word sense disambiguation. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management. pp. 2317–2320.
OpenNLP, 2014. The Apache Software Foundation. Apache OpenNLP. [Online]. <http://opennlp.apache.org> (accessed 02.16).
Patwardhan, S., Pedersen, T., 2006. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In: Proc. EACL 2006 Workshop on Making Sense of Sense: Bringing Computational Linguistics and Psycholinguistics Together. Trento, Italy, pp. 1–8.
Preiss, J., Stevenson, M., 2013. Unsupervised domain tuning to improve word sense disambiguation. In: Proc. 2013 Conference of the North American Chapter of the Association for Computational Linguistics. pp. 680–684.
Reddy, S., Inumella, A., McCarthy, D., Mark, S., 2010. IIITH: domain specific word sense disambiguation. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 387–391.
Reisinger, J., Mooney, R.J., 2010. Multi-prototype vector-space models of word meaning. In: Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 109–117.
Resnik, P., 1995. Using information content to evalutate semantic similarity in a taxonomy. In: 14th International Joint Conference on Artificial Intelligence. Montreal, Quebec, Canada, pp. 448–453.

Richardson, R., Smeaton, A.F., Murphy, J., 1994. Using WordNet as a knowledge base for measuring semantic similarity between words. In: Proceedings Conference on Artificial Intelligence and Cognitive Science.

Sánchez, D., Batet, M., Isern, D., Valls, A., 2012. Ontology-based semantic similarity: a new feature-based approach. Expert Syst. Appl. 39 (9), 7718–7728.

Schutz, A., Buitelaar, P., 2005. Relext: a tool for relation extraction from text in ontology extension. In: *Proc.* International Semantic Web Conference ISWC'05. Galway, Ireland, pp. 593–606.

Selvi, P., Gopalan, N.P., 2007. Measuring semantic similarity between the concepts based on an ontology. Int. J. Soft Comput. 2 (5), 617–623.

Seng, Y., Ng, H., 2007. Domain adaptation with active learning for word sense disambiguation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. pp. 49–56.

Serra, I., Girardi, R., Novais, P., 2014. Evaluating techniques for learning non-taxonomic relationships of ontologies from text. Expert Syst. Appl. 41 (11), 5201–5211.

Siddharth, P., Pedersen, T., 2003. *The CPAN WordNet::Similarity*. [Online]. <http://wn-similarity.sourceforge.net> (accessed 02.16).

Sinha, R., Mihalcea, R., 2007. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In: Proceedings of the International Conference on Semantic Computing. pp. 363–369.

Snyder, B., Palmer, M., 2004. The English all-words task. In: Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text. Barcelona, Spain, pp. 41–43.

Tejada-Cárcamo, J., Calvo, H., Gelbukh, A., Hara, K., 2010. Unsupervised WSD by finding the predominant sense using context as a dynamic thesaurus. J. Comp.Sci. Technol. 25 (5), 1030–1039.

Terra, E.L., Clarke, C.L.A., 2003. Frequency estimates for statistical word similarity measures. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1. pp. 165–172.

Tika, 2014. The Apache Software Foundation. Apache Tika. [Online]. <http://tika.apache.org> (accessed 02.16).

Toutanova, K., Klein, D., Manning, C.D., Singer, Y., 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings NAACL '03 Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1. pp. 173–180.

Weeds, J., Weir, D., 2005. Co-occurrence retrieval: a flexible framework for lexical distributional similarity. Comput. Ling. 31 (4), 439–475.

Wróblewska, A., Protaziuk, G., Bembenik, R., Podsiadly-Marczykowska, T., 2013. Associations between texts and ontology. In: Intelligent Tools for Building a Scientific Information Platform. pp. 305–321.

Zhao, S., Lin, D., 2005. A nearest-neighbor method for resolving PP-Attachment ambiguity. In: Proceedings of the First International Joint Conference on Natural Language Processing. pp. 545–554.