International Journal of Computational Linguistics and Applications, vol. 5, no. 2, 2014, pp. 67–94
Received 14/05/2014, accepted 12/08/2014, final 27/10/2014.
ISSN 0976-0962, http://ijcla.bahripublications.com

Discriminative Ability of WordNet Senses on the Task of Detecting Lexical Functions in Spanish Verb-Noun Collocations

OLGA KOLESNIKOVA

Instituto Politécnico Nacional, Mexico

ABSTRACT

Collocations, or restricted lexical co-occurrence, are a difficult issue in natural language processing because their semantics cannot be derived from the semantics of their constituents. Therefore, such verb-noun combinations as "take a break," "catch a bus," "have lunch" can be interpreted incorrectly by automatic semantic analysis. Since collocations are combinations frequently used in texts, errors in their analysis cannot be ignored. The quality of analysis of collocations can be improved if they are annotated with lexical functions that represent semantic classes of collocations. In this work, we study how WordNet senses viewed as sets of hypernyms can distinguish lexical functions of Spanish verb-noun collocations in experiments with supervised machine learning methods. We show that WordNet senses discriminate lexical functions to different degrees depending on the function, and this phenomenon can be used to evaluate the quality of word sense definitions as well as to measure similarity of various senses of a word and the correlation between word senses and lexical functions.

1 INTRODUCTION

Collocation is a word combination whose semantics cannot be derived from typical meaning of each component word. Very often collocations are combinations of two words. For example, *have* is commonly interpreted as 'possess', and the meaning of *lunch* is 'midday meal'. However, *have lunch* cannot be understood as 'possess a midday meal' since the noun *lunch* preserves its typical meaning of a midday meal, but the verb *have* acquires another meaning, 'consume'. Therefore, *have lunch* is correctly interpreted as 'consume a midday meal'. Due to this peculiarity of the verb-noun combination *have lunch*, it is termed collocation to distinguish it from other syntactically similar phrases termed free word combinations whose meaning can be represented as a sum of meanings of their component words: *have a daughter*, *have a book*, *have a nice house*, etc.

Recognition and correct interpretation of collocations is a big challenge in natural language processing (NLP). Errors in semantic analysis of collocations cannot be easily ignored due to their high frequency: about 43% of entries in the English WordNet are collocations [c, d]; also, depending on a specific domain, collocations can comprise up to 85% of vocabulary in texts [11]. Therefore, adequate detection and adequate processing of collocations plays a very significant role in all natural language processing applications that include a module for performing semantic analysis of texts to various degrees of granularity.

As previously mentioned, in the verb-noun collocation *have lunch*, the noun *lunch* preserves its typical sense, but the verb *have* changes its meaning. Why is it so? It seems from the set of synonyms of *have* which is {*command*, *enjoy*, *hold*, *own*, *retain*} (taken from Merriam-Webster Thesaurus online, http://www.merriam-webster.com), *lunch* chooses a combination with *have* in order to generate the meaning 'consume food in the afternoon'. Notice that the noun *food* prefers another verb, *take*, to express the same semantics of 'consuming a solid substance used for nourishment'. So, two different verbs *have* and *take* express the same meaning but each of them in combination with different nouns. Such usage is also termed 'restricted lexical co-occurrence' meaning that we can say *have lunch* but not **take lunch* in the sense of eating it. Therefore, in a collocation, one word "chooses" another one; in the case of verb-noun collocations, a noun chooses a verb and modifies its meaning. The noun is called the base of a collocation, and the verb is called the collocate. In this paper we study only verb-noun collocations.

There are many state-of-the-art methods for automatic detection and extraction of collocations. Such techniques produce lists of collocations. Lists of collocations would be more useful if collocations were tagged with semantic information. In other words, semantic analysis of collocations is needed in order to interpret their meaning correctly.

In some cases such collocations correspond to so-called concepts [14]. In sentiment analysis and opinion mining it is very important to identify such concepts [15, 21]. In standard sentiment lexicons, collocations are either ignored or assigned neutral polarity. However, modern research shows that collocations carry meaning and sentiment. They play major role in, for example, contextual polarity shifting [3]. Collocations are also useful to understand emotions [4]. Some researchers use concept vectors instead of bag-of-words models [2, 19].

In this paper, we consider semantic analysis of a certain type, namely, semantic classification of collocation according to lexical functions. This task can also be viewed as automatic detection of lexical functions in collocations. The concept of lexical function is a formalism within the Meaning-Text Theory [7, 8] and is explained in the section that follows (Section 2, Lexical Functions).

In this paper, we study how and with what precision lexical functions can be distinguished by WordNet senses [9]. WordNet senses can be characterized by many features included in this ontology: glosses (definitions of words), synsets (words synonymous to a given word), relations (hypernymy, hyponymy, antonymy, meronymy, troponymy, entailment), sentence frames, examples of word usage, etc. We represent each word sense by a set of all its hypernyms. A hypernym is a word whose meaning is more generic than the meaning of a given word; for example, *furniture* is a hypernym of *chair*.

We chose the hypernymy relation to represent word senses because this feature has been used in the state of the art research on automatic detection of lexical functions, so there is enough experimental data published in literature to compare our results with. Also, as it will be seen in the next section, lexical function is a tool designed to generalize semantics of collocations, so supposedly hypernyms as words with more general semantics can be helpful in lexical function identification.

In this research we analyze the ability of hypernyms of verb-noun collocation constituents to discriminate lexical functions. What hypernyms and corresponding to what word senses discern lexical functions with a higher precision? This is the basic issue we deal with here. Also, we consider another issue: how variations in lexical function detection precision depending on WordNet senses can be understood and interpreted with respect to such characteristic of collocations as relative or non-compositionality [26].

For automatic detection of lexical functions, we use a dataset of Spanish verb-noun collocations [5, 23] and supervised machine learning techniques.

The rest of the paper is organized as follows. Section 2 discusses the concept of lexical functions as semantic classes of collocations. Section 3 shows some applications of lexical functions in natural language processing. In Section 4 we review state of the art research on automatic detection of lexical functions. In Section 5 we define the problem and questions we deal with in this research. Section 6 describes our experiments, their results are discussed in Section 7, and Section 8 presents conclusions and outlines future work.

2 LEXICAL FUNCTIONS

Lexical function (LF) is a formal concept proposed within the Meaning-Text Theory [7, 8] to generalize and represent both semantic and syntactic structure of a collocation. LF is similar to a mathematical function and has the form

$$\mathrm{LF}(w_0) = \{w_1, w_2, \ldots, w_n\}, \qquad (1)$$

where w_0 is the LF argument which is the base of a collocation, and the LF value is the set $\{w_1, w_2, \ldots, w_n\}$ whose elements are words or word combinations $w_i, 0 < i \leq n$, which is/are collocate/s of a given base. In the present research we consider only verb-noun collocations and, respectively, verb-noun lexical functions, so applying the above formula to this particular group of collocations we have w_0 to denote a noun (base of a collocation) and the set $\{w_1, w_2, \ldots, w_n\}$ will now include only one element w_1 which is a verb (collocate in a verb-noun collocation). Thus we will study lexical functions of the following type:

$$\mathrm{LF}\colon \mathrm{N} \to V, \qquad (2)$$

where *N* is a set of nouns in which each noun functions as a base in a verb-noun collocation, and *V* is a set of all verb collocates.

LF in the Formulas 1 and 2 represents the generalized semantics of groups of verbal collocates on the one hand, and on the other hand, captures the basic syntactic and predicate-argument structure of sentences in which a collocation belonging to such group is used. Therefore, a lexical function can be viewed as a formal representation of semantic, syntactic, and governing patterns in which verb-noun collocations participate. We will explain and illustrate this formalism with some of most common lexical functions:

– Oper1, from Latin *operari = do, carry out*, formalizes the action of carrying out of what is denoted by the noun (LF argument). Integers in the LF notation are used to specify the predicate-argument and syntactic structure. In Oper1, 1 means that the word used to lexicalize the semantic role of agent of the action denoted by the verb (agent is considered the first argument of a verb) functions as the grammatical subject in a sentence, so Oper1 represents the pattern *Agent performs* w_0 (w_0 is the argument of a lexical function, see Formula 1). For example, Oper1(*decision*) = *make*, and in the sentence *The president made a decision*, *president* is the agent and its syntactic function is subject. Other verb-noun collocations which can be covered by Oper1 are *pursue a goal*, *make an error*, *apply a measure*, *give a smile*, *take a walk*, *have lunch*, *deliver a lecture*, *make an announcement*, *lend support*, *put up resistance*, *give an order*.
– Oper2 has the meaning 'undergo, meet' and represents the pattern *Patient undergoes* w_0, for example, *suffer a change*, *receive support*, *receive an order*, *meet resistance*.
– Func0, from Latin *functionare = to function*, represents the meaning 'happen, take place'. The noun argument w_0 of Func0 is the name of an action, activity, state, property, relation, i.e., it is such a noun whose meaning is or includes a predicate in the logical sense of the term thus presupposing arguments. Zero in Func0 means that the argument of Func0 is the agent of the verb and functions as the grammatical subject in a sentence. Therefore, Func0 represents the patterns w_0 *occurs*. For example, *snow falls, silence reigns, smell lingers, time flies*.
– Real1, from Latin *realis* = *real*, means 'to use the noun argument w_0 according to its destination', 'to do with w_0 what one is supposed to with w_0', 'to do with regard to w_0 what is normally expected of the agent', so Real1 represents the pattern *Agent acts according to* w_0: *do one's duty*, *fulfill an obligation*, *keep a secret*, *follow a principle*, *obey a command*.

Each lexical function discussed above represents one simple meaning or a single semantic unit, so such functions are called simple. There are lexical functions that formalize combinations of unitary meanings; they are called complex lexical functions. Now we will consider some of them:

– IncepOper1 is a combination of the semantic unit 'begin', from Latin *incipere*, and Oper1 presented above. This LF has the meaning 'begin doing something' and represents the pattern *Agent begins to do the* <*noun*>: *to open fire on …*, *to acquire popularity*, *to sink into despair*, *to take an attitude*, *to obtain a position*, *begin negotiations*, *fall into problems*.
– ContOper1 combines the meaning 'continue', from Latin *continuare*, with Oper1. It represents the pattern *Agent continues to do* w_0, for example, *maintain enthusiasm*, *maintain supremacy*, *keep one's balance*.
– Caus, from Latin *causare*, represents the meaning 'cause, do something so that w_0 begins occurring'. Caus is used only in combinations with other LFs. So CausFunc0 means 'to cause the existence of w_0' and represents the pattern *Agent does something such that* w_0 *begins to occur*: *bring about the crisis*, *create a difficulty*, *present a difficulty*, *call elections*, *establish a system*, *produce an effect*. CausFunc1 represents the pattern *Non-agent argument does something such that* w_0 *begins to occur*, for example, *open a perspective*, *raise hope*, *open a way*, *cause a damage*, *instill a habit into somebody*.

3 APPLICATION OF LEXICAL FUNCTIONS IN NATURAL LANGUAGE PROCESSING

Lexical functions possess a number of important properties which make them an effective tool for natural language processing. First, LFs are universal; it means that a significantly little number of LFs (about 70) represent the fundamental semantic relations between words in the vocabulary of any natural language and the basic semantic relations which syntactically connected word forms can obtain in a text. Secondly, LFs are characteristic for idioms in many natural languages and can serve as a typology for classification of idioms, collocations, and other types of restricted lexical co-occurrence. Thirdly, LFs can be paraphrased. For example, the LFs Oper and Func can form combinations with their arguments which are synonymous to the basic verb like in the following utterances: *The government controls prices – The government has control of prices –* The *government keeps prices under control – The prices are under the government's control.*

LFs can be used to resolve syntactic ambiguity. In such cases, syntactically identical phrases are characterized by different lexical functions which serve as a tool for disambiguation.

For example, consider two phrases: *support of the parliament* and *support of the president.* In the first phrase *support* is the object, but in the second phrase *support* functions syntactically as the subject and semantically as the agent. The surface phrase structure in both cases is identical: *support* + *of* + noun; this fact causes syntactic ambiguity and due to it both phrases may have both meanings: 'support given by the parliament (by the president)', which syntactically is the subject interpretation with the agentive syntactic relation between *support* and the subordinated noun, and 'support given to the parliament (to the president)' which syntactically is the object interpretation with the first completive syntactic relation between *support* and the subordinated noun. This type of ambiguity is often extremely difficult to resolve, even within a broad context. LF verbs can be successfully used to disambiguate such phrases because they impose strong limitations on the syntactic behavior of their arguments in texts.

Now let us view the same phrases in a broader context. The first example is *The president spoke in support of the parliament,* where the verb *to speak in* is Oper1 of the noun *support,* i.e., Oper1(*support*) = *speak in*. Oper1 represents the pattern *Agent performs* w_0 (where w_0 is the argument of Oper1), so *the president* is interpreted as the agent, and *support* as the object. Therefore, *the president spoke in support of the parliament* can only be interpreted as describing the support given to the parliament, with *parliament* having the syntactic function of the complement of *support*.

On the other hand, verbs of Oper2 participate in another pattern: *Patient undergoes* w_0. So Oper2 verb is by definition a verb whose grammatical subject represents the patient of w_0 and in the utterance *the president enjoyed* (Oper2) *the support of the parliament,* the phrase *the support of the parliament* implies the support given to the president by the parliament, with *parliament* having the syntactic function of the agentive dependent of the noun *support*.

LFs can also be used in computer-assisted language learning. It is a well-known fact in second language teaching practice that collocations are difficult to master by learners, so learner's speech often sounds unnatural due to errors in restricted lexical co-occurrence. To deal with this issue, a lexical function dictionary can be used whose advantage is that it includes the linguistic material on word combinations which is absent in word dictionaries.

LFs can be used in machine translation due to their semantic universality and cross-linguistic idiomaticity. These characteristics make LFs an ideal tool for selecting idiomatic translations of set expressions in a machine translation system. *They took a walk after lunch* is translated into Spanish by Google Translate as *Tomaron un paseo después del almuerzo* (translated on May 6, 2015). In English, Oper1(*walk*) = *take*, but in Spanish Oper1 of the argument *paseo* (English *walk*) is *dar* (English lit. *give*). So Oper1(*paseo*) = *dar*, however, the system translated the collocation *take a walk* literally as *tomar paseo*, since *take* is literally *tomar* in Spanish. Therefore, a module that annotates word combinations with lexical functions can be included in any machine translation system to improve the quality of translation of collocations and idiomatic expressions.

Patterns corresponding to LFs can be used in other natural language processing tasks: parsing, semantic role tagging, text analysis, etc. For example, LF patterns can be used as templates for generating grammatical sentences in automatic text generation.

4 Automatic Detection of Lexical Functions

There have been made a few attempts to detect LFs automatically. Wanner [28] approached automatic detection of LFs as a task of automatic classification of collocations according to LF typology. He applied Nearest Neighbor machine learning technique to classify Spanish verb-noun pairs according to nine LFs selected for the experiments. The distance of candidate instances to instances in the training set was evaluated using path length in hypernym hierarchy of the Spanish part of EuroWordNet [25, 27] corresponding to each verb and noun. An average F-measure of about 70% was achieved in these experiments. The largest training set (for CausFunc0) included 38 verb-noun pairs and all test sets had the size of 15 instances.

Alonso Ramos *et al*. [1] proposed an algorithm for extracting collocations following the pattern *support verb + object* from the FrameNet corpus of examples [22] and checking if they are of the type Oper. This work takes advantage of syntactic, semantic, and collocation annotations in the FrameNet corpus, since some annotations can serve as indicators of a particular LF. The authors tested the proposed algorithm on a set of 208 instances. The algorithm showed an accuracy of 76%. Alonso Ramos *et al*. conclude that extraction and semantic classification of collocations is feasible with semantically annotated corpora. This statement sounds logical because the formalism of lexical function captures the correspondence between the semantic valence of the keyword and the syntactic structure of utterances where the keyword is used in a collocation together with the value of the respective LF.

5 PROBLEM AND RESEARCH QUESTIONS

Wanner in [28] and also we in our previous work [5] interpreted the task of LF detection as a task of classification of verb-noun collocations into two classes: collocations which belong to a particular LF and those which do not belong to this LF. To classify verb-noun collocations, both works applied supervised machine learning methods. In the training set supplied to machine learning algorithms, hypernyms of both verb and noun of each collocation extracted from the Spanish WordNet [25, 27] were used as features. In order to retrieve hypernyms, all words in the training set of verb-noun collocations were annotated with Spanish WordNet senses as well as with their respective LFs.

In the experiments in [28], LFs were detected with an F-measure of about 70% which can be considered sufficiently well however not excellent. The author of [28] analyzes the reasons of classification errors and concludes that two of them are caused by limitations of the Spanish WordNet. Firstly, some senses are absent in this lexical resource: for example, in *observar la costumbre* (lit. observe the custom) *observar* means *follow*, *keep*; however, this sense is absent in the Spanish WordNet. Secondly, some descriptions in the Spanish WordNet are imprecise: for example, semantic descriptions of *periodico* (newspaper) and *libro* (book) differ from each other to a great extent in spite of the fact that these two words are similar; for a more detailed discussion of imprecise descriptions see [28].

In our experiments reported in [5] we did not include in the dataset those collocations in which the verb and/or the noun do not have their proper senses in the Spanish WordNet. Nevertheless, the performance of supervised classifiers with 10-fold cross validation on the training set did not improve a lot: we obtained an F-measure of about 73%.

It is obvious that some classification errors are due to faults of the supervised learning methods themselves, but we suppose that another obstacle can be found in an insufficient ability of some verb sense definitions to distinguish the semantics of lexical functions, i.e., the meanings of verbs they acquire in collocations. So our hypothesis is that in spite of the fact that such verb sense definitions do represent the meanings of verbs in collocations, the quality of such representation in the part of hypernyms corresponding to such definitions in some cases is not sufficient for discriminating lexical functions.

We mentioned in the Introduction that in a verb-noun collocation, the noun, as the base of the collocation, is used in its typical sense, though the verb, being the collocate and thus semantically dependent on the noun, is not used in its typical meaning but the noun imposes another meaning on the verb. In this work we want to study the correlation between the quality of verb definitions viewed as sets of respective hypernyms and the ability of machine learning methods to discriminate among lexical functions.

Consequently, in this research we intend to respond to the following research questions:

1. To what measurable degree does the meaning of the verb in a collocation differ from the typical meaning of the same verb?
2. Is such degree the same or different for different lexical functions?
3. Is the WordNet sense (represented as a set of hypernyms) which corresponds to the meaning of the verb in a collocation able to distinguish lexical functions and if yes to what degree?
4. How can we measure the correlation between lexical functions and WordNet senses?

To find answers to these questions, we designed three types of experiments with a dataset of Spanish verb-noun collocations annotated with lexical functions [5, 23] and senses of the Spanish WordNet version 2000611 [25, 27]. In the experiments of all types we used supervised machine learning techniques and hypernyms of both the verb and the noun in a collocation as features, including the verb and the noun themselves as zero-level hypernyms.

1. Experiments on the whole dataset as in our previous work [5]. We repeated these experiments since we use a different number of examples for some lexical functions and a more recent version of Weka thus aiming at a more adequate comparison of the results of these experiments with the results of the other experiments in this work.
2. Experiments on only such collocations of the dataset in which the verb has a sense other than 1. Commonly, a list of senses in WordNet is ordered by frequency, so sense 1 is the most frequent meaning of a word which can be considered as its typical meaning. Thus, the training set in this kind of experiments includes collocations in which the meaning of the verb differs from its typical meaning. (Here we have to remark, that for some collocations, the meaning of the verb in a collocation is most frequently met in corpora and thus is put as sense 1. In such a case, most frequent does not mean most typical. However, what meaning should be considered typical is another research issue; here for our purposes we will adopt the interpretation of typical as most frequent.)
3. Experiments on the same collocations as in the experiments of type 2, but for each verb, we change its sense to sense 1.

To put it simpler in the text that follows, we use Experiment 1, Experiment 2, and Experiment 3 to refer to experiments of type 1, 2, and 3, respectively.

6 EXPERIMENTS

6.1 *Experiment 1*

In Experiment 1 on automatic detection of lexical functions, we used a dataset of Spanish most frequent lexical verb-noun functions [5, 23] compiled by manually annotating each word with the Spanish WordNet [25, 27] senses, and each verb-noun pair as a particular LF or FWC (free word combination). Verb-noun pairs in the dataset are the first 1000 samples in a list of verb-noun pairs retrieved from the Spanish Web

Corpus with 116,900,060 tokens [6, 24] and ordered by frequency. Table 1 presents the statistics of our LF dataset.

Table 1. Lexical functions found in 1000 most frequent verb-noun pairs in the Spanish Web Corpus. For each LF, the number of instances (#) is given as well as their total frequency (Freq) in the corpus; FWC is free word combination (verb-noun pair which is not a collocation)

LF	#	Freq
Oper1	280	165319
FWC	202	70211
CausFunc1	90	45688
CausFunc0	112	40717
Real1	61	19191
Func0	25	17393
IncepOper1	25	11805
Oper2	30	8967
Caus2Func1	16	8242
ContOper1	16	5354
Manif	13	3339
Copul	9	2345
CausPlusFunc0	7	2203
Func1	4	1848
PerfOper1	4	1736
CausPlusFunc1	5	1548
Real2	3	1547
FinOper1	6	1476

LF	#	Freq
PerfFunc0	1	1293
Caus1Oper1	2	1280
Caus1Func1	3	1085
IncepFunc0	3	1052
PermOper1	3	910
CausManifFunc0	2	788
CausMinusFunc0	3	746
Oper3	1	520
LiquFunc0	2	514
IncepReal1	2	437
Real3	1	381
PlusOper1	1	370
CausPerfFunc0	1	290
AntiReal3	1	284
MinusReal1	1	265
AntiPermOper1	1	258
ManifFunc0	1	240
CausMinusFunc1	1	229
FinFunc0	1	178

In Section 2, we did not explain the meaning of all LFs presented in Table 1, only the meaning of the most frequent ones. Definitions and examples for the rest of LFs in Table 1 can be consulted in [8].

An interesting fact can be observed in Table 1: the frequency of verb-noun collocations tagged as Oper1 is higher than the frequency of free verb-noun combinations (FWC). This fact re-affirms the significance of a correct analysis and interpretation of collocations in automatic processing of texts in natural languages.

For our experiments, we chose the first eight LFs in Table 1. Note that FWC stands for free word combinations which are not considered as belonging to lexical functions. The first eight LFs have a sufficient number of samples which allows their usage in supervised machine learning techniques. However, as a training set we used all samples of the data set only excluding two types of examples. To the first type belong erroneous instances which were retrieved automatically due to parser errors, for example, combinations containing non-letter symbols. The second type of samples which we excluded from the training set are such for whose verb and/or noun the Spanish WordNet does not have an appropriate sense, such deficiency of this widely used dictionary was mentioned in [28].

After removing the latter two types of samples from the dataset, our training set included 900 verb-noun combinations. Table 2 presents the eight LFs we experimented with and their respective number of instances, and Table 3 gives examples of each LF in Table 2.

Table 2. Lexical functions used in Experiment 1

LF	# of instances
Oper1	266
Oper2	28
IncepOper1	24
ContOper1	16
Real1	60
Func0	16
CausFunc0	109
CausFunc1	89
Total	608

We applied supervised machine learning algorithms implemented in Weka 3-6-12-x64 [29, 30] to classify each sample in the training set as belonging to a particular LF or not (binary yes-no classification) using 10-fold cross validation. Each sample was represented as a set of all hypernyms of the verb and all hypernyms of the noun including the verb and the noun as zero-level hypernyms. Hypernyms were retrieved from the Spanish WordNet.

As mentioned in Section 5, such experiments were performed by us in previous work and reported in [5]. However, we considered it necessary to repeat the same experiments, first of all, due to the fact that here we use a more recent version of Weka and, for some LFs, a different number of samples in the training set than in [5]. Secondly, we intend to compare the results of our previous experiments in [5] with Experiments 2 and 3 performed in this research. To make a fair and adequate comparison we will have all the experiments done with the same implementation version of machine learning algorithms and on the same training set. Finally, in this paper we will give a more extended report of the results of Experiment 1 than for the same type of experimentation performed in [5].

Table 3. Examples of lexical functions from our the training set

LF	Examples of collocations	
	Spanish	English translation
Oper1	*realizar un estudio* *cometer un error* *dar un beso*	do a study make an error give a kiss
Oper2	*recibir tratamiento* *obtener una respuesta* *sufrir daño*	receive treatement get an answer suffer a damage
IncepOper1	*iniciar un proceso* *tomar la palabra* *adoptar la actitud*	begin a process take the floor adopt the attitude
ContOper1	*seguir un curso* *mantener un contacto* *guardar silencio*	follow a course keep in touch keep silent
Real1	*satisfacer una necesidad* *lograr un objetivo* *resolver un conflicto*	satisfy a need reach a goal resolve a conflict
Func0	*el tiempo pasa* *una posibilidad cabe* *la razón existe*	time flies there is a possibility there exists a reason
CausFunc0	*crear una cuenta* *formar un grupo* *hacer ruido*	create an account form a group make noise
CausFunc1	*ofrecer una posibilidad* *causar un problema* *crear una condición*	offer a possibility cause a problem create a condition

Tables 4–7 present the results of the experiments described in [5] but conducted now as we explained above. In the results, we included the best 10 classifiers in terms of F-measure for each of the eight lexical functions given in Table 2.

The overall average best F-measure for eight lexical functions used in Experiment 1 is 0.734 or about 73%. The work of Wanner [28] reviewed in Section 4 reports an average F-measure of about 70% in two experiments on detection of the following lexical functions: Oper1, Oper2, ContOper1, CausFunc0, Caus2Func1, IncepFunc1, FunFunc1, Real1, and Real2. Our result of 73% shows a slight improvement compared with [28]; however, such comparison is not fair since we did not experiment with all LFs and the same set of LF instances as in [28].

Table 4. Ten best classifiers on detection of Oper1 and Oper2, respectively

Oper1	
Classifier	F-m
trees.SimpleCart	0.879
rules.PART	0.873
trees.BFTree	0.872
bayes.Bayesian LogisticRegression	0.868
meta.Attribute SelectedClassifier	0.867
meta.Bagging	0.867
trees.LADTree	0.866
meta.END	0.865
meta.FilteredClassifier	0.865
meta.Ordinal ClassClassifier	0.865
Average best	0.869

Oper2	
Classifier	F-m
functions.SimpleLogistic	0.739
meta.LogitBoost	0.739
rules.DecisionTable	0.723
meta.Bagging	0.711
meta.Attribute SelectedClassifier	0.708
meta.END	0.708
meta.FilteredClassifier	0.708
meta.Ordinal ClassClassifier	0.708
trees.J48	0.708
trees.LADTree	0.708
Average best	0.716

Table 5. Ten best classifiers on detection of IncepOper1 and ContOper1, respectively

IncepOper1	
Classifier	F-m
rules.Prism	0.732
trees.FT	0.711
bayes.Bayesian LogisticRegression	0.700
functions.SMO	0.683
misc.VFI	0.682
rules.Nnge	0.682
trees.LADTree	0.682
meta.RandomCommittee	0.667
trees.Id3	0.650
meta.Attribute SelectedClassifier	0.619
Average	0.681

ContOper1	
Classifier	F-m
lazy.LWL	0.800
rules.DecisionTable	0.800
trees.REPTree	0.800
trees.Id3	0.788
meta.Attribute SelectedClassifier	0.774
rules.Ridor	0.774
trees.BFTree	0.774
trees.SimpleCart	0.774
meta.END	0.750
meta.FilteredClassifier	0.750
Average	0.778

Table 6. Ten best classifiers on detection of Real1and Func0, respectively

Real1	
Classifier	F-m
meta.LogitBoost	0.667
meta.Bagging	0.660
trees.BFTree	0.660
functions.SMO	0.649
rules.Jrip	0.647
rules.Nnge	0.635
trees.LADTree	0.634
trees.FT	0.627
bayes.Bayesian LogisticRegression	0.624
trees.REPTree	0.611
Average best	0.641

Func0	
Classifier	F-m
meta.Attribute SelectedClassifier	0.824
rules.Jrip	0.824
trees.ADTree	0.800
meta.END	0.788
meta.FilteredClassifier	0.788
meta.Ordinal ClassClassifier	0.788
rules.PART	0.788
rules.Ridor	0.788
trees.BFTree	0.788
trees.J48	0.788
Average best	0.796

Table 7. Ten best classifiers on detection of CausFunc0 and CausFunc1, respectively

CausFunc0	
Classifier	F-m
rules.Jrip	0.722
trees.LADTree	0.712
trees.SimpleCart	0.710
trees.BFTree	0.705
trees.REPTree	0.704
meta.Bagging	0.679
trees.FT	0.678
functions.SMO	0.676
trees.ADTree	0.670
bayes.Bayesian LogisticRegression	0.664
Average best	0.692

CausFunc1	
Classifier	F-m
meta.RotationForest	0.744
meta.END	0.732
meta.FilteredClassifier	0.732
meta.Ordinal ClassClassifier	0.732
rules.DecisionTable	0.732
trees.J48	0.732
rules.Jrip	0.729
trees.BFTree	0.727
meta.LogitBoost	0.718
trees.LADTree	0.718
Average best	0.730

Another state of the art paper by Alonso Ramos *et al.* [1] surveyed in Section 4 as well reported an accuracy of 76% on extraction of verb-noun collocations of the type Oper from the FrameNet corpus of examples [22]. Here we will mention that our average F-measure on detection of Oper1 and Oper2 is 0.793 or 79%.

6.2 *Experiments 2 and 3*

In the training set for Experiment 2, we included only such collocations of the original dataset (used in Experiment 1) in which the verb has a sense other than 1, that is, the verb has a sense other than its typical sense. In Experiment 3, the number of each verb sense in the training set used in Experiment 2 is substituted by 1. The purpose of this substitution is to compare the performance of classifiers on LF detection using actual (non-typical) verb senses against the performance of the same classifiers on the same collocations using sense 1 (typical) of the verbs.

We believe that comparison of results of these two experiments will shed light on the research questions posed in Section 5.

Table 8 shows, for each lexical function, the total number of instances (i.e., verb-noun collocations), the number of instances in which the verb has sense 1, and the number of collocations in which the verb has sense other than 1, the latter verb-noun pairs were used in Experiments 2 and 3.

Table 8. Lexical functions

LF	Total # of instances (used in Experiment 1)	# of instances with sense 1 of the verb	# of instances with sense ≠ 1 of the verb (used in Experiments 2 and 3)
Oper1	266	112	154
Oper2	28	22	6
CausFunc0	109	29	80
CausFunc1	89	16	73
IncepOper1	24	3	21
ContOper1	16	2	14
Real1	60	44	16
Func0	16	6	10
FWC	196	123	73

The methodology and procedures applied in these experiments are the same as in Experiment 1: in the training set, each verb-noun collocations is represented as a set of hypernyms of the verb and the noun, and the training set was submitted to all applicable to this data type supervised learning methods implemented in Weka 3-6-12-x64 [29, 30].

7 RESULTS AND DISCUSSION

The results of ten best classifiers in Experiment 1 are presented in Section 6.1, see Tables 4-7. However, in this section we refer to the performance of some classifiers in Experiment 1 which did not appear among the best ten ones, but since their performance is high in Experiment 2, we present their values of F-measure in Experiment 1 to make a comparison with their performance in Experiments 2 and 3.

Tables 9–12 show the results of all three experiments. We arranged the results in a way convenient for comparison. The tables include the results for each of the eight lexical functions in the format as follows.

The first column contains ten best classifiers in Experiment 2 ordered by performance on the training set in which the verbs have meanings other than 1, i.e., their actual meaning. The respective F-measure for each classifier is given in the second column entitled $\neq$ 1 (Exp.2). The third column entitled 1(Exp.3) contains F-measure of the same classifiers applied to the same training set, but in which each verb is assigned sense 1 (Experiment 3). The fourth column entitled (Exp.2)–(Exp.3) includes the difference between two values: F-measure for the case of the verb sense other than 1 and F-measure for the case of substitution of the actual verb sense with sense 1 (the difference between the results of Experiment 2 and Experiment 3). The fifth column gives the values of F-measure for the classifiers in the first column they reached in Experiment 1, where these classifiers were applied to the original dataset described in Section 6.1. For each of the four columns with the values of F-measure, average is given as well.

Now we will discuss the results of the experiments for each lexical function. It can be observed in Table 9 that Oper1 is detected with almost the same F-measure on the whole dataset and on the set with verb senses other than 1 (0.866 and 0.899, respectively). However, when we substituted verb senses other than 1 with sense 1, the performance became notably worse, with an F-measure of 0.808. This observation suggests that actual verb senses fit well the definition of Oper1, *Agent performs* w_0*,* where w_0 is the noun in a verb-noun collocation.

For example, consider an Oper1 collocation *realizar*_6 *estudio*_5, lit. realize a study; the numbers here are the Spanish WordNet senses. *Realizar*_6 belongs to the synset {*efectuar*_1, *realizar*_6, *llevar_a_cabo*_5, *hacer*_15}, lit. effect, realize, accomplish, do, and its hypernym is the synset {*actuar*_2, *hacer*_6} (lit. act, do). On the other hand, *realizar*_1 is in the synset {*causar*_1, *realizar*_1, *crear*_1}, lit. cause, realize, create, which has no hypernym. Clearly, *realizar* in *realizar estudio* does not mean *cause* or *create*, therefore, sense 6 of *realizar* is an adequate correspondence to the meaning of this verb in the collocation under consideration. It seems that for the other Oper1 verbs in the dataset, the situation is the same or very similar.

Table 9. Experimental results for Oper1 and Oper2

Oper1

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
trees.SimpleCart	0.900	0.815	0.085	0.879
rules.PART	0.900	0.783	0.117	0.873
trees.LADTree	0.900	0.769	0.131	0.866
meta.END	0.900	0.832	0.068	0.865
meta.FilteredClassifier	0.900	0.832	0.068	0.865
meta.OrdinalClassClassifier	0.900	0.832	0.068	0.865
trees.J48	0.900	0.832	0.068	0.865
rules.Jrip	0.900	0.819	0.081	0.857
trees.BFTree	0.897	0.819	0.078	0.872
trees.REPTree	0.896	0.750	0.146	0.854
Average	0.899	0.808	0.091	0.866

Oper2

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
functions.SimpleLogistic	0.800	0.800	0	0.739
meta.AttributeSelectedClassifier	0.800	0.800	0	0.708
meta.END	0.800	0.800	0	0.708
meta.FilteredClassifier	0.800	0.800	0	0.708
meta.OrdinalClassClassifier	0.800	0.800	0	0.708
rules.DecisionTable	0.800	0.800	0	0.723
rules.Jrip	0.800	0.800	0	0.696
rules.OneR	0.800	0.800	0	0.619
rules.PART	0.800	0.800	0	0.681
trees.BFTree	0.800	0.667	0.133	0.694
Average	0.800	0.787	0.013	0.698

The results for Oper2 in Table 9 are quite different from those for Oper1. While for Oper1 the actual verb senses distinguish well the semantics of this function, it can be observed that Oper2 is distinguished with the same F-measure by the actual verb senses (other than 1) and by sense 1: nine out of ten best classifiers show the same value of F-measure (0.800) in Experiment 2 and Experiment 3.

Certainly, the results for Oper2 are by no means representative of this whole semantic class of verb-noun collocations since we used a dataset with a very small number of positive examples (6 collocations of Oper2 as positive examples and the rest 872 instances as negative examples). However, we submitted this set to the classifiers in order to make some observations that might sketch lines of future research.

The almost equal performance of the classifiers in Experiment 2 and Experiment 3 on Oper2 detection can be explained by some faults in sense definitions given in the Spanish WordNet. As an example, let us consider *sufrir*_3 *cambio*_3, lit. suffer a change. *Sufrir*_3 belongs to the synset {*soportar*_3, *sufrir*_3}, lit. bear, suffer, and has the following two synsets as hypernyms: {*experimentar*_3}, lit. experience, and {*actuar*_2, *llevar_a_cabo*_3, *hacer*_8}, lit. act, accomplish, do.

On the other hand, *sufrir*_1 belongs to the synset {*aguantar*_4, *tolerar*_1, *sufrir*_1, *soportar*_2}, lit. endure, tolerate, suffer, bear, which has one hypernym {*dejar*_2, *permitir*_2}, lit. allow, permit. Although *experience* is the verb with a clear Oper2 semantics, however, it may be considered too general to classify its hyponym *sufrir* (suffer) as a value of Oper2 for *cambio* (change) as an argument. On the contrary, the verbs *endure*, *tolerate*, *bear*, *allow*, *permit* in combination with the noun *change* have a less general and more specific meaning of *undergo* (*a change*) thus serving as better features for Oper2 detection.

The above example also illustrates the fact that in some cases it is not easy to find the most appropriate word sense for a given lexical function. In Table 9 we see as well that the classifier performance on all samples of Oper2 is worse (F-measure=0.698) than on those samples of Oper2 in which the verb has sense other than 1 (F-measure=0.800). We believe that due to the prevalence of verb sense 1 in the dataset for Oper2 (22 examples with verb sense 1 of total 28 examples, see Table 8), the performance on this dataset is lower.

Similar differences among the results of the three experiments considered for Oper2 in the previous paragraph are also observed for Real1 and Func0, see Table 11.

Table 10 presents the results for IncepOper1. Here the performance of classifiers in terms of F-measure on the dataset with verb senses other than 1 is significantly higher than the classifier performance on the whole dataset (0.812 against 0.644). However, if we substitute sense other than 1 with sense 1, the performance degrades dramatically (0.812 for verb senses other than 1 versus 0.644 for verb sense 1). It may mean that sense 1 introduces noise into the set of features used for classification, and the other senses communicate the semantics of IncepOper1 more precisely.

Table 10. Experimental results for IncepOper1 and ContOper1

IncepOper1

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
trees.Id3	0.900	0.571	0.329	0.650
rules.Prism	0.842	0.583	0.259	0.732
rules.Nnge	0.829	0.516	0.313	0.682
trees.LADTree	0.829	0.629	0.200	0.682
functions.SMO	0.821	0.571	0.250	0.683
functions.Logistic	0.810	0.435	0.375	0.569
meta.MultiClassClassifier	0.810	0.435	0.375	0.567
BayesianLogisticRegression	0.789	0.286	0.503	0.700
functions.SimpleLogistic	0.789	0.429	0.360	0.556
rules.PART	0.703	0.439	0.264	0.615
Average	0.812	0.489	0.323	0.644

ContOper1

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
lazy.LWL	0.857	0.880	–0.023	0.800
rules.DecisionTable	0.857	0.923	–0.066	0.800
functions.SimpleLogistic	0.857	0.923	–0.066	0.733
BayesianLogisticRegression	0.857	0.923	–0.066	0.714
rules.Ridor	0.839	0.963	-0.124	0.774
meta.AttributeSelectedClassifier	0.828	0.889	–0.061	0.774
trees.BFTree	0.828	0.889	–0.061	0.774
trees.SimpleCart	0.828	0.923	–0.095	0.774
meta.END	0.828	0.889	–0.061	0.750
meta.FilteredClassifier	0.828	0.889	–0.061	0.750
Average	0.841	0.909	–0.068	0.764

For example, consider the collocation *tomar*_6 *poder*_1, lit. take the power. *Tomar*_6 belongs to the synset {*asumir*_2, *tomar*_6}, lit. assume, take, and has a hypernym synset {*comenzar*_7, *iniciar*_7, *empezar*_6}, lit. commence, initiate, begin. Let us compare the latter with sense 1 of *tomar*: it is in the synset {*conseguir*_1, *tomar*_1, *sacar*_1, *obtener*_1}, lit. get, take, receive, obtain, and has no hypernym, therefore, first, its meaning is most general in this branch of the Spanish WordNet graph, and secondly, its meaning is very different from the semantics of *tomar* in *tomar poder*. The same is true for other IncepOper1 collocations: observe that most of them have verb senses other than 1 (21 out of 24, see Table 8) which represent the IncepOper1 semantics sufficiently well.

Table 11. Experimental results for Real1 and Func0

Real1

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
trees.LADTree	0.692	0.667	0.025	0.634
rules.Prism	0.643	0.818	–0.175	0.608
functions.SMO	0.621	0.593	0.028	0.649
rules.Nnge	0.621	0.609	0.012	0.635
trees.FT	0.621	0.720	–0.099	0.627
meta.END	0.609	0.545	0.064	0.606
meta.FilteredClassifier	0.609	0.545	0.064	0.606
meta.OrdinalClassClassifier	0.609	0.545	0.064	0.606
trees.J48	0.609	0.545	0.064	0.606
rules.PART	0.609	0.545	0.064	0.602
Average	0.624	0.613	0.011	0.618

Func0

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
meta.AttributeSelectedClassifier	0.857	0.783	0.074	0.824
rules.Jrip	0.857	0.720	0.137	0.824
trees.ADTree	0.857	0.900	–0.043	0.800
meta.END	0.857	0.526	0.331	0.788
meta.FilteredClassifier	0.857	0.526	0.331	0.788
meta.OrdinalClassClassifier	0.857	0.526	0.331	0.788
rules.PART	0.857	0.750	0.107	0.788
trees.J48	0.857	0.526	0.331	0.788
trees.REPTree	0.857	0.571	0.286	0.788
functions.SMO	0.857	0.857	0	0.778
Average	0.857	0.668	0.189	0.795

Results for ContOper1 in Table 10 are surprising. If the verb senses other than 1 (which pretend to be the actual ones according to a human expert) are changed to sense 1, this improves the classifier performance.

Experiment 2 (verb senses $\neq$ 1) showed an average F – measure of 0.841, but Experiment 3 (verb senses $\neq$ 1 substituted with 1) showed an average F-measure of 0.909. The classifier performance on the whole dataset in Experiment 1 is poorer with an average F-measure of only 0.764. It seems that the semantics of ContOper1 *continue to do what is denoted by the noun* is expressed as verbs' typical sense, i.e., sense 1.

Table 12. Experimental results for CausFunc0 and CausFunc1

CausFunc0

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2)| –(Exp.3)	Exp.1
trees.SimpleCart	0.756	0.532	0.224	0.710
trees.LADTree	0.744	0.744	0	0.712
meta.AttributeSelectedClassifier	0.735	0.829	–0.094	0.649
trees.BFTree	0.726	0.818	–0.092	0.705
functions.SimpleLogistic	0.714	0.812	–0.098	0.633
meta.END	0.711	0.829	–0.118	0.628
meta.FilteredClassifier	0.711	0.829	–0.118	0.628
meta.OrdinalClassClassifier	0.711	0.829	–0.118	0.628
trees.J48	0.711	0.769	–0.058	0.628
rules.Jrip	0.704	0.843	–0.139	0.722
Average	0.722	0.783	–0.061	0.664

CausFunc1

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
meta.RotationForest	0.771	0.855	–0.153	0.744
trees.BFTree	0.771	0.870	–0.157	0.727
trees.SimpleCart	0.771	0.859	–0.163	0.711
meta.END	0.769	0.861	–0.164	0.732
meta.FilteredClassifier	0.769	0.861	–0.193	0.732
meta.OrdinalClassClassifier	0.769	0.861	–0.167	0.732
trees.J48	0.769	0.861	–0.191	0.732
meta.LogitBoost	0.766	0.892	–0.205	0.718
trees.ADTree	0.766	0.892	–0.254	0.636
meta.AttributeSelectedClassifier	0.762	0.892	–0.145	0.705
Average	0.768	0.870	–0.179	0.717

But what features of sense 1 influence the performance of the classifiers? Let us, as an example, consider *llevar*_5 *vida*_5, lit. spend life. In this collocation, *llevar*_5 does not have synonyms, and its hypernym is {*usar*_2}, lit. use, so *llevar vida* is interpreted as *use life* in the meaning *continue to live a life*, and this interpretation is correct.

On the other hand, *llevar*_1 belongs to the synset {*acarrear*_1, *traer*_1, *llevar*_1, *transportar*_1}, lit. carry, bring, take, transport, and its hypernym is {*cargar*_1, *transportar*_2, *desplazar*_1, *mover*_1}, lit. bear, transport, displace, move. At the first sight, *move* has nothing to do with the semantics of *spend* in *spend life*. We can note here, that *use* in *use life* implies a process, therefore, continuing to do something; however, *use* fails to serve as an umbrella semantic representation of *continue* for all ContOper1 verbs such as *mantener* (maintain), *seguir* (follow), *guardar* (keep), etc. Since *continue* implies movement or transition from one state to another, such words as *displace*, *move* have a better coverage of ContOper1 verbs and their degree of generalization is sufficient for detecting ContOper1 in verb-noun collocations.

The same phenomenon is observed in detection of CausFunc0 and CausFunc1: the classifier performance is improved if verb senses other than 1 in Experiment 2 are substituted with verb sense 1 in Experiment 3. For CausFunc0, average values of F-measure are 0.722 in Experiment 2 and 0.783 in Experiment 3, and for CausFunc1, 0.768 and 0.870 in Experiments 2 and 3, respectively. Similarly to what was said in the previous paragraph we can say that verbs used in the meaning of CausFunc0 are distinguished better with their typical senses. These are such verbs as *abrir* (open), *agregar* (add), *alcanzar* (reach), *aportar* (contribute), *aprobar* (approve), *causar* (cause), *construir* (construct), *convocar* (call), *crear* (create), *dar* (give), *declarar* (declare), *dejar* (allow), *desarrollar* (develop), *elaborar* (elaborate), *escribir* (write), *establecer* (establish), *formar* (form), *hacer* (do), *introducir* (introduce), *poner* (put), *producir* (produce), *proporcionar* (provide), etc. The same can be said about CausFunc1 verbs: *abrir* (open), *causar* (cause), *constituir* (construct), *crear* (create), *dar* (give), *dejar* (allow), *despertar* (wake), *destacar* (*highlight*), *establecer* (*establish*), *hacer* (do), *ofrecer* (offer), *poner* (put), *prestar* (lend), *producir* (produce), *proporcionar* (provide), *reservar* (reserve). It can be observed from the examples of the verbs, that the same verbs are used in both functions; this explains why the same phenomena of a better performance for verb sense 1 is observed for both functions.

Now let us consider the research questions we posed in Section 5. Firstly, the difference between the classifier performance in

Experiment 2 and Experiment 3 can bear evidence of the degree of similarity / dissimilarity between the typical sense of a verb in a verb-noun collocation and the verb's meaning in the collocation. We observed that such degree of similarity is higher for CausFunc1 (-0.179), ContOper1 (-0.068) and CausFunc0 (-0.061). The lower degree have Real1 (0.011), Oper2 (0.013), Oper1 (0.091), Func0 (0.189), and IncepOper1 (0.323). Secondly, we observed that this similarity degree varies among lexical functions. Thirdly, the results of Experiment 1 and Experiment 2 as well as the similarity degree just mentioned show to what extent the meaning of the verb in a collocation is able to distinguish lexical functions. Lastly, the difference between the classifier performance in Experiment 2 and Experiment 3 can serve also as a measure of correlation between lexical functions and Spanish WordNet senses which also can be used to evaluate the quality of word sense definitions.

8 CONCLUSIONS AND FUTURE WORK

In this work, we have studied to what degree WordNet senses can distinguish among semantic classes of verb-noun collocations represented as lexical functions, a concept of the Meaning-Text Theory by I. Mel'čuk proposed in order to generalize the semantics of restricted lexical co-occurrence or collocations.

We have experimented with supervised machine learning methods on a dataset of Spanish verb-noun collocations annotated with lexical functions and the Spanish WordNet senses. Lexical functions represent such concepts as *do* (what is denoted by the noun), *undergo*, *begin to do*, *continue to do*, etc. Each concept covers a large group of verb-noun collocations thus representing various semantic classes of collocations. Detection of each lexical function was performed as a binary classification using hypernyms of verbs and nouns as features.

We have observed that 5 of 8 lexical functions chosen for the experiments were discriminated well by the actual verb senses with which a human expert annotated them; an average F-measure showed by classifiers on these 5 lexical functions was 0.798. However, 3 of 8 lexical functions were better discerned by classifiers if the actual verb sense was substituted by sense 1, in this case an average F-measure was 0.854 against 0.777 for the case of the actual verb senses of the same lexical functions.

We considered some factors which could cause this phenomenon including imprecise word definitions in WordNet as well as a too high level of generalization of hypernyms. On the other hand, the difference in the performance of classifiers on detection of lexical functions depending on the WordNet senses can be used to measure similarity of senses as well as correlation between semantic classes of verb-noun collocations and WordNet senses; it can also be used to evaluate the quality and discriminative ability of WordNet senses.

In future, we plan to perform a more detailed and extensive analysis of the results obtained in the experiments reported in this work. We also plan to analyze the role of spotting collocations for different text analysis tasks, such as textual entailment [1213], sentiment analysis [16], emotion detection [18, 20], and personality recognition [17], among others.

ACKNOWLEDGEMENTS We are grateful to Vojtěch Kovář and other members of the Sketch Engine team, www.sketchengine.co.uk, for providing us a list of most frequent verb-noun pairs from the Spanish Web Corpus. The work was done under partial support of Mexican Government: SNI, BEIFI-IPN, and SIP-IPN grants 20144534, 20152095.

REFERENCES

1. Alonso Ramos, M., Rambow O., Wanner L.: Using semantically annotated corpora to build collocation resources. Proceedings of LREC, Marrakesh, Morocco, pp. 1154–1158 (2008).
2. Cambria, E., Fu, J., Bisio, F., Poria, S.: AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis. In: Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 508–514 (2015)
3. Cambria, E., Poria, S., Gelbukh, A., Kwok, K.: Sentic API: A common-sense based API for concept-level sentiment analysis. In: #Microposts2014, 4th Workshop on Making Sense of Microposts at WWW 2014, CEUR Workshop Proceedings, vol. 1141, pp. 19–24 (2014)
4. Das, D., Poria, S., Bandyopadhyay, S.: A classifier based approach to emotion lexicon construction. In Natural Language Processing and Information Systems, Springer, pp. 320–326 (2012)
5. Gelbukh, A., Kolesnikova, O.: Supervised learning for semantic classification of Spanish collocations. Advances in Pattern Recognition, Springer Berlin Heidelberg, pp. 362–371 (2010).
6. Kilgarriff, A., Rychly, P., Smrz, P., Tugwell, D.: The Sketch Engine. Proceedings of EURALEX 2004, pp. 105–116 (2004)

7. Mel'čuk, I. A.: Lexical Functions: A Tool for the Description of Lexical Relations in a Lexicon. In: Wanner, L. (ed) Lexical Functions in Lexicography and Natural Language Processing, Benjamins Academic Publishers, Amsterdam, Philadelphia, PA, pp. 37–102 (1996)
8. Mel'čuk, I. A., Žolkovskij, A. K.: Towards a functioning 'Meaning-Text' model of language. Linguistics, vol. 8(57), pp. 10–47 (1970)
9. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM, vol. 38(11), pp. 39–41 (1995)
10. Miller, G. A., Leacock, C., Tengi, R., Bunker, R. T.: A semantic concordance. In Proceedings of the workshop on Human Language Technology Association for Computational Linguistics, pp. 303–308 (1993)
11. Nakagawa, H., Mori, T.: Automatic term recognition based on statistics of compound nouns and their components. Terminology, vol. 9(2), pp. 201–219 (2003)
12. Pakray, P., Pal, S., Poria, S., Bandyopadhyay, S., Gelbukh, A.: JU_CSE_TAC: Textual entailment recognition system at TAC RTE-6. In System Report, Text Analysis Conference Recognizing Textual Entailment Track (TAC RTE) Notebook (2010)
13. Pakray, P., Poria, S., Bandyopadhyay, S., Gelbukh, A.: Semantic textual entailment recognition using UNL. Polibits, vol. 43, pp. 23-27 (2011)
14. Poria, S., Agarwal, B., Gelbukh, A., Hussain, A., Howard, N.: Dependency-based semantic parsing for concept-level text analysis. In: Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2014. Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 8403, Springer, pp. 113–127 (2014)
15. Poria, S., Cambria, E., Ku, L. W., Gui, C., Gelbukh, A.: A rule-based approach to aspect extraction from product reviews. In: Proceedings of the Second Workshop on Natural Language Processing for Social Media, SocialNLP 2014, pp. 28–37 (2014)
16. Poria, S., Cambria, E., Winterstein, G., Huang, G.B.: Sentic patterns: Dependency-based rules for concept-level sentiment analysis. Knowledge-Based Systems, vol. 69, pp. 45–63 (2014)
17. Poria, S., Gelbukh, A., Agarwal, B., Cambria, E., Howard, N.: Common sense knowledge based personality recognition from text. In: Advances in Soft Computing and Its Applications, MICAI 2013, Lecture Notes in Artificial Intelligence, vol. 8266, Springer, pp. 484-496 (2013)
18. Poria, S., Gelbukh, A., Cambria, E., Das, D., Bandyopadhyay, S.: Enriching SenticNet polarity scores through semi-supervised fuzzy clustering. In: Workshop on Sentiment Elicitation from Natural Text for Information Retrieval and Extraction, SENTIRE 2012, 2012 IEEE 12th International Conference on Data Mining Workshops (ICDMW), IEEE CS Press, pp. 709-716 (2012)

19. Poria, S., Gelbukh, A., Cambria, E., Hussain, A., Huang, G.B.: EmoSenticSpace: A novel framework for affective common-sense reasoning. Knowledge-Based Systems, vol. 69, pp. 108-123 (2014)
20. Poria, S., Gelbukh, A., Hussain, A., Howard, N., Das, D., Bandyopadhyay, S.: Enhanced SenticNet with affective labels for concept-based opinion mining. IEEE Intelligent Systems, vol. 28, no. 2, 31-38 (2013)
21. Poria, S., Ofek, N., Gelbukh, A., Hussain, A., Rokach, L.: Dependency tree-based rules for concept-level aspect-based sentiment analysis. In: Semantic Web Evaluation Challenge, SemWebEval 2014 at ESWC 2014, Greece, Revised Selected Papers. Communications in Computer and Information Science, vol. 475, Springer, pp. 41-47 (2014)
22. Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C. R., Scheffczyk, J.: FrameNet II: Extended Theory and Practice, http://framenet.icsi.berkeley.edu/book/book.pdf/ ICSI Berkeley (2006)
23. Spanish Verb-Noun Lexical Functions, free download at http://148.204.58.221/okolesnikova/index.php?id=lex or http://www.gelbukh.com/lexical-functions
24. Spanish Web Corpus in SketchEngine, http://trac.sketchengine.co.uk/wiki/Corpora/SpanishWebCorpus/
25. Spanish WordNet, http://www.lsi.upc.edu/~nlp/web/index.php?Itemid=57&id=31&option=com_content&task=view
26. Svensson, M. H.: A very complex criterion of fixedness: Non-compositionality. In: Granger, S. Meunier, F. (eds.) Phraseology: an interdisciplinary perspective, Amsterdam & Philadelphia: John Benjamins Publishing Company, pp. 81–93 (2008)
27. Vossen P. (ed): EuroWordNet: A Multilingual Database with Lexical Semantic Networks, Kluwer Academic Publishers, Dordrecht (1998)
28. Wanner, L.: Towards automatic fine-grained classification of verb-noun collocations. Natural Language Engineering, vol. 10(2), pp. 95–143. (2004)
29. Witten, I. H., Frank, E., Hall, M. A.: Data mining: Practical machine learning tools and techniques. Morgan Kaufmann Publishers, MA, USA (2011)
30. The University of Waikato Computer Science Department Machine Learning Group, WEKA download, http://www.cs.waikato.ac.nz/ml/weka/downloading.html

OLGA KOLESNIKOVA
SUPERIOR SCHOOL OF COMPUTER SCIENCES,
INSTITUTO POLITÉCNICO NACIONAL (IPN)
MEXICO CITY, 07738, MEXICO
E-MAIL: <KOLESOLGA@GMAIL.COM >