Using WEKA for Semantic Classification of Spanish Verb-Noun Collocations

Olga Kolesnikova and Alexander Gelbukh

Center for Computing Research, National Polytechnic Institute
Mexico City, 07738, Mexico
kolesolga@gmail.com, www.gelbukh.com

Abstract. Collocations, or restricted lexical co-occurrence, can be classified according to their semantics represented by the formalism of lexical functions in the frame of Meaning-Text Theory. We use learning algorithms implemented in WEKA to classify Spanish verb-noun collocations according to lexical functions. Experiments were made on a manually built corpus of verb-noun pairs. WEKA classifiers were tested for detection of two lexical functions, namely Oper1 and CausFunc0. Some WEKA classifiers show better performance than state-of-the-art results.

1 Introduction

Knowledge of collocation is important for natural language processing because collocation comprises the restrictions on how words can be used together. Automatic extraction techniques produce a list of collocations. Such lists are more valuable if collocational data is tagged with semantic information. In this section, we will consider the notion of collocation, approaches to classification of collocations, and lexical functions as semantic typology for collocational classification.

1.1 Collocations

Collocation has been a controversial issue in linguistic research for a number of decades. Many attempts have been made to define collocation. These definitions are based on various criteria: statistical, lexical, functional, structural, semantic. We make use of a semantic definition of collocation given in [9]. This definition affirms that collocation is a combination of two lexical items in which the semantics of one of the lexical items (called the "base" of collocation) is autonomous from the combination it appears in, and where the other lexical item (called the "collocate") adds semantic features to the semantics of the base. In other words, collocation is a binary word combination in which the base is used in its "normal" meaning and the collocate is used in a "non-typical" meaning. As an example, consider a Spanish collocation *dar un paseo* '*give a walk*'. The noun *paseo* is the base and is used in its typical meaning, but the verb *dar* here means '*do*' although it's most frequent meaning is '*give*'. There are other verb-noun collocations where the verbal collocate acquires the same

© M. Martínez, A. Alarcón (Eds.)
Advances in Computer Science and Engineering.
Research Computing Science 45, 2010, pp. 221-232

Received 30/04/10
Final version 19/05/10
Accepted 14/05/10

meaning ‘*do*’: *cometer un error* (‘*make a mistake*’), *ejercer control* ‘*exercise control*’, *hacer una pregunta* ‘*ask a question*’, *realizar el esfuerzo* ‘*make an effort*’. We can observe that all these collocations can be characterized by the meaning pattern ‘*do something*’. The next group of collocations reveals another meaning pattern – ‘*cause something to exist*’: *causar daño* ‘*cause damage*’, *abrir una posibilidad* ‘*open a posibility*’, *establecer una regla* ‘*establish a rule*’, *formar un grupo* ‘*form a group*’. Therefore, semantic patterns ‘*do something*’, ‘*cause something to exist*’ represent semantic contents of respective collocations.

1.2 Classification of collocations

Collocations are classified on the basis of various criteria. According to their grammatical structure, collocations can be grammatical and lexical [5]. Depending on the part of speech of the base and the collocate, collocations can be adjective-noun, verb-noun, adverb-adjective, verb-adverb, etc. [3]. If frequency of collocational parts in corpus is applied as a classification principle, then ‘upward’ and ‘downward’ collocation are distinguished by Sinclair [13]. Wanner proposed a semantic classification of collocation based on the taxonomy of lexical functions [17].

1.3 Lexical function

Lexical function (LF) is a formalism developed within the Meaning-Text Theory [8] [10] to represent semantic and syntactic structure of collocation. It has a general form

$$\mathrm{LFn}_1\ldots\mathrm{n}_k(b) = c,$$

where b is the base of a collocation, c is the collocate. In terms of the Meaning-Text Theory, b is called the keyword, and c – the LF value. “LF” in the formula stands for the name of lexical function. The LF name is an abbreviated Latin word which denotes the semantic contents of collocations. The string of positive integers “$\mathrm{n}_1\ldots\mathrm{n}_k$“ is optional. The integer value indicates semantic valency of the keyword. The position of the integer in the string represents the syntactic function of the word used to fill in the corresponding semantic valency (first position signifies the subject, second – direct object, etc). Let us consider a few examples.

Oper1(*paseo*) = *dar*. The keyword (the base of the collocation) is *paseo, ‘walk’*. The LF value is *dar*, ‘*give*’. The lexical function has the name “Oper” from Latin *operari* – ‘*do, carry out*’. Since *paseo* denotes an action, it can be viewed in its verbal aspect. In this case, its first semantic role is the agent. The integer “1” means that the word used to lexicalize the role of agent, functions as a subject in a sentence, for example *El doctor salió a dar un paseo por la tarde.* ‘*The doctor left to take a walk in the afternoon*’, where *the doctor* is the agent.

Func0(*posibilidad*) = *existir*. “Func” is from Latin *functionare* – to ‘*function*’, so respective collocations have meaning ‘*something functions, happens, takes place*’. The integer “0” means that no semantic role is lexicalized as subject, but the keyword itself functions as the subject: *No existe la posibilidad de realizar este proyecto.* ‘*No possibility exists to realize this project*’.

Manif1(*problema*) = *plantear*. "Manif" is from Latin *manifestare* – to '*manifest*'. Respective collocations mean that the agent of the verb reveals something that it becomes apparent. *Plantear problema* in Spanish corresponds to *pose a problem* in English.

The above examples demonstrate so-called simple lexical functions which formalize a single semantic element, or one meaning like '*do*', '*function*', '*manifest*'. However, there are many cases when collocations have more complex meaning which is formed by a combination of two or more "single" meanings. This phenomenon is captured by complex LFs. Before giving examples of complex LF, it should be mentioned that there exist simple LFs that are seldom used independently but more often constitute parts of complex LFs. Table 1 lists names of such simple LFs and their respective meanings taken from [10].

Table 1. Examples of simple LFs used in complex LFs more often than independently.

LF	**Meaning**	**Comment**
Incep	Lat. *incipere* – '*begin*'	something begins occurring
Cont	Lat. *continuare* – '*continue*'	something continues occurring
Fin	Lat. *finire* – '*cease*'	something ceases occurring
Caus	Lat. *causare* – '*cause*'	do something so that a situation begins occurring
Liqu	Lat. *liquidare* – '*liquidate*'	do something so that a situation stops occurring

A complex LF is a combination of two or more simple LFs. Table 2 presents a few complex LFs and their meanings, for each complex LF examples of collocations are presented. "K" in the column "Meaning" stands for the keyword.

Table 2. Examples of complex LFs.

LF	**Meaning**	**Keyword**	**LF Value**	**Collocations**
IncepOper1	begin to do K	*proceso*	*iniciar*	*iniciar el proceso*
		'process'	*'begin'*	*'begin the process'*
		responsabilidad	*asumir*	*asumir la responsabilidad*
		'responsibility'	*'assume'*	*'assume the responsibility'*
ContOper1	continue to do K	*contacto*	*mantener*	*mantener el contacto*
		'contact'	*'maintain'*	*'maintain the contact'*
		camino	*seguir*	*seguir el camino*
		'road'	*'follow'*	*'follow the road'*
CausFunc0	cause that K comes into existence	*efecto*	*producir*	*producir el efecto*
		'effect'	*'produce'*	*'produce the effect'*
		explicación	*dar*	*dar una explicación*
		'explanation'	*'give'*	*'give an explanation'*
LiquFunc0	do something that K ceases to exist	*vida*	*quitar*	*quitar la vida*
		'life'	*'take away'*	*'take (one's) life'*
		problema	*evitar*	*evitar el problema*
		'problem'	*'avoid'*	*'avoid the problem'*

As mentioned before, collocations are classified on the basis of various criteria. Since LFs represent semantic patterns of collocations, LF taxonomy can be used to build a semantic classification of collocations. Besides, the taxonomy of LFs is advantageous because it groups collocations according to language-independent generalized semantics and characteristic syntactic patterns. Implemented in a computer readable dictionary of collocations, a classification by lexical functions will allow effective use of collocations in natural language applications including parsers, high quality machine translation, systems of paraphrasing and computer-aided learning of lexica [2].

The rest of the paper is organized as follows. Section 2 summarizes previous research on automatic detection of LFs. Section 3 defines the objective of this work. We discuss the experimental results in Section 4. Section 5 presents conclusions and outlines future work.

2 Related work

2.1 Automatic Detection of Lexical Functions

There have been made a few attempts to detect LFs automatically. Wanner approached automatic detection of LFs as the task of automatic classification of collocations according to LF typology [17]. 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 hyperonym hierarchy of the Spanish part of EuroWordNet [16]. An average f-score of about 70% was achieved in these experiments. The largest training set included 38 verb-noun pairs (for LF CausFunc0) and all test sets had the size of 15 instances.

Alonso Ramos *et al.* [1] propose an algorithm for extracting collocations following the pattern “support verb + object” from FrameNet corpus of examples [12] and checking if they are of the type Oper*n*. 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 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 valency 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.

2.2 Collocation classification according to LF

Wanner *et al.* [18] experiment with the same type of lexical data as [17], i.e. verb-noun pairs. The task is to answer the question: what kind of collocational features are fundamental for human distinguishing among collocational types. The authors view collocational types as LFs, i.e. a particular LF represents a certain type of collocations. Three hypotheses are put forward as possible solutions, and to model every solution, an appropriate machine learning technique is selected. Below we list the three hypotheses and the selected machine learning techniques.

1. Collocations can be recognized by their similarity to the prototypical sample of each collocational type; this strategy is modeled by the Nearest Neighbor technique.
2. Collocations can be recognized by similarity of semantic features of their elements (i.e., base and collocate) to semantic features of elements of the collocations known to belong to a specific LF; this method is modeled by Naïve Bayesian network and a decision tree classification technique based on the ID3-algorithm.
3. Collocations can be recognized by correlation between semantic features of collocational elements; this approach is modeled by Tree-Augmented Network Classification technique.

In classification experiments, the authors deal with two groups of verb-noun collocations. The first group includes only verb-noun pairs where nouns belong to the semantic field of emotions. In the second group, nouns are field-independent. We will compare the results for the second group of collocations with WEKA performance in Section 4.3. It should be mentioned also, that having proposed three hypotheses, the authors have not yet demonstrated their validity by comparing the performance of many machine learning techniques known today, but apply only four learning algorithms to illustrate that three human strategies mentioned above are practical. This will be considered in more detail in Section 4.3.

3 Objective

The aim of this paper is to test WEKA methods for classification of collocations according to LFs. We apply WEKA classifiers to a corpus of Spanish verb-noun combinations and train the system to detect two lexical functions chosen for the experiments. For each LF in question, the classifier with best results is identified. The obtained results are compared with those in [18] for verb-noun collocations with field-independent nouns.

4 Experimental results

4.1 Experimental Methodology

We apply machine learning techniques as implemented in the WEKA version 3-6-2 learning and data mining toolset [6] [21]. The data as described in Section 4.2 was supplied to 67 classifiers of various classes. We evaluated the performance of WEKA classifiers by comparing precision, recall, and *f*-measure (weighted average values over all classes). The precision is the proportion of the examples which truly have class *x* among all those which were classified as class *x*. The recall is the proportion of examples which were classified as class *x*, among all examples which truly have class *x*. The *f*-measure is 2*Precision*Recall/(Precision+Recall) [19].

4.2. Data

For compiling data sets, we used a list of verb-object pairs extracted by the Word Sketch Engine [7] from Spanish web corpus [14] and ranked by frequency. 1000 most frequent pairs were annotated with part of speech, Spanish WordNet version 200611 [16] [15] word senses, and LFs by human experts.

Table 3 gives the number of samples in the list for all LFs found in this list. The number of instances that were annotated as free word combinations was 198; 61 pairs were qualified as errors (for example, some pairs contained symbols like “, ©, -- instead of words, other pairs turned out to be combinations “verb + past participle”).

Table 3. LFs with the respective number of samples for 1000 most frequent verb-noun pairs in Spanish Web Corpus.

LF	Number of samples	LF	Number of samples
Oper1	157	Real2	3
CausFunc0	102	IncepFunc0	3
CausFunc1	60	MinusCausFunc0	3
Real1	45	ManifCausFunc0	2
Func0	22	LiquFunc0	2
Oper2	16	AntiReal3	1
IncepOper1	14	Oper3	1
ContOper1	11	PlusCausFunc0	1
Copul	9	FinOper1	1
Manif	9	MinusCausFunc1	1
Caus2Func1	9	FinFunc0	1
PlusCausFunc0	6	PermOper1	1
PlusCausFunc1	5	Real3	1
PerfOper1	4	Caus1Oper1	1
Caus1Func1	3	PerfFunc0	1

Since the most frequent LFs in our data were Oper1 and CausFunc0, we constructed one data set for Oper1 and another set for CausFunc0. These are the steps we followed to build a training set for each LF out of the source file which contained the list of 1000 most frequent pairs annotated with POS, word senses, and LFs as described above.

1. Samples of the LF in question are marked as positive examples and are included in the training set.
2. Verb-noun pairs of all other LFs are marked as negative examples and are included in the training set. Pairs qualified as errors are not made a part of the training set. Free word combinations are considered a lexical function and are incorporated into the training set as negative examples.
3. All hyperonyms for every word in the set of positive and negative examples are extracted from the Spanish WordNet. Synsets to which the words in the examples belong are considered zero-level hyperonyms.
4. The training set for WEKA tool has the Attribute-Relation File Format (ARFF) [22]. Every hyperonym in the set of Step 3 is considered as a nominal attribute which can take one of two values: “1” if it is a hyperonym of any word in a given

verb-noun pair, and “0” if it is not. The overall number of attributes in the training set as it was supplied to WEKA included 935 attributes, 934 of them were hyperonyms. The last attribute was the class attribute with two possible values: “yes” for positive examples and “no” for negative ones. Therefore, every verb-noun pair was represented as a vector of length 935.

An example of the input file in ARFF format is given in Fig. 1. The record “@data” marks the beginning of the data section. All records beginning with “@attribute” as well as all strings in the data section are accompanied by a comment starting with symbol “%” in ARFF. For attribute entries, comment includes words of the synset specified by its number after the record “@attribute”. For data strings, comment includes the verb-noun pair represented by this string. The pair is annotated with POS and word senses. Comments were added for human viewing of data.

Partial representation of AFRR file

```
@relation Oper1

@attribute n00001740 {0,1} % entidad_1
@attribute n00002086 {0,1} % ser_vivo_1 ser_1 organismo_1
@attribute n00003731 {0,1} % agente_causal_1 causa_4
...
...
@attribute v01128460 {0,1} % causar_5 producir_4 ocasionar_1
@attribute v01130277 {0,1} % hacer_2
@attribute v01131926 {0,1} % dar_9
...
...
@attribute category {yes,no}

@data
1,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,...,0,0,0,yes % v_hacer_15 n_mención_1
...
...
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,...,0,0,0,no % v_abrir_5 n_camino_5
```

4.3 Results and Discussion

Training sets constructed as described in Section 4.2 were supplied to 67 classifiers. The classifiers were tested by ten-fold cross validation. The obtained results are presented in Table 4, Table 5, and Table 6.

Table 4. Performance of WEKA Classifiers of class lazy on Oper1 and CausFunc0 data sets, P stands for precision, R – for recall, F – for *f*-measure.

Classifier class	Classifier	Oper1			CausFunc0		
		P	R	F	P	R	F
lazy	IB1	0.777	0.744	0.753	0.853	0.869	0.859
	IBk	0.776	0.728	0.739	0.855	0.865	0.859
	KStar	0.772	0.752	0.759	0.858	0.872	0.864

Table 5. Performance of WEKA Classifiers of classes bayes, function, meta, and rules on Oper1 and CausFunc0 data sets, P stands for precision, R – for recall, F – for *f*-measure.

Classifier class	Classifier	Oper1			CausFunc0		
		P	R	F	P	R	F
bayes	AODE	0.841	0.845	0.842	0.883	0.89	0.856
	AODEsr	0.802	0.808	0.803	0.852	0.83	0.84
	BayesianLogisticRegression	**0.927**	**0.927**	**0.927**	0.901	0.907	0.903
	BayesNet	0.836	0.832	0.834	0.88	0.883	0.881
	HNB	0.857	0.859	0.852	0.845	0.877	0.837
	NaiveBayes	0.837	0.84	0.838	0.861	0.885	0.858
	NaiveBayesSimple	0.837	0.84	0.838	0.861	0.885	0.858
	NaiveBayesUpdateable	0.837	0.84	0.838	0.861	0.885	0.858
	WAODE	0.838	0.84	0.839	0.883	0.897	0.881
functions	LibSVM	0.507	0.712	0.593	0.765	0.875	0.816
	Logistic	0.913	0.905	0.907	0.9	0.89	0.894
	RBFNetwork	0.828	0.833	0.828	0.854	0.88	0.842
	SimpleLogistic	**0.92**	**0.92**	**0.92**	0.9	0.907	0.902
	SMO	**0.922**	**0.922**	**0.922**	0.911	0.915	0.913
	VotedPerceptron	0.894	0.896	0.894	0.88	0.893	0.883
	Winnow	0.677	0.687	0.681	0.812	0.764	0.785
meta	AdaBoostM1	0.828	0.808	0.777	0.829	0.875	0.819
	AttributeSelectedClassifier	0.913	0.913	0.913	0.912	0.919	0.913
	Bagging	0.913	0.913	0.913	0.917	0.922	0.918
	ClassificationViaClustering	0.599	0.666	0.616	0.772	0.774	0.773
	ClassificationViaRegression	0.894	0.893	0.894	**0.923**	**0.923**	**0.923**
	CVParameterSelection	0.507	0.712	0.593	0.765	0.875	0.816
	Dagging	0.89	0.889	0.889	0.895	0.905	0.887
	Decorate	0.891	0.887	0.889	0.896	0.899	0.898
	END	0.91	0.91	0.91	0.909	0.916	0.91
	EnsembleSelection	0.917	0.917	0.917	0.926	0.929	0.927
	FilteredClassifier	0.91	0.91	0.91	0.909	0.916	0.91
	Grading	0.507	0.712	0.593	0.765	0.875	0.816
	LogitBoost	0.914	0.915	0.914	0.898	0.907	0.899
	MultiBoostAB	0.819	0.769	0.709	0.765	0.875	0.816
	MultiClassClassifier	0.913	0.905	0.907	0.9	0.89	0.894
	MultiScheme	0.507	0.712	0.593	0.765	0.875	0.816
	OrdinalClassClassifier	0.91	0.91	0.91	0.909	0.916	0.91
	RacedIncrementalLogitBoost	0.507	0.712	0.593	0.765	0.875	0.816
	RandomCommittee	0.874	0.868	0.87	0.895	0.902	0.898
	RandomSubSpace	0.879	0.875	0.867	0.884	0.897	0.877
	RotationForest	0.904	0.905	0.904	0.908	0.915	0.909
	Stacking	0.507	0.712	0.593	0.765	0.875	0.816
	StackingC	0.507	0.712	0.593	0.765	0.875	0.816
	ThresholdSelector	0.916	0.915	0.915	0.887	0.895	0.89
	Vote	0.507	0.712	0.593	0.765	0.875	0.816
rules	ConjunctiveRule	0.784	0.762	0.705	0.765	0.875	0.816
	DecisionTable	0.905	0.906	0.905	0.902	0.907	0.904
	JRip	0.914	0.915	0.914	**0.932**	**0.933**	**0.932**
	NNge	0.894	0.892	0.893	0.888	0.895	0.891
	OneR	0.819	0.769	0.709	0.877	0.893	0.871
	PART	0.911	0.912	0.911	0.896	0.899	0.897
	Prism	0.881	0.874	0.876	0.912	0.909	0.911
	Ridor	0.895	0.896	0.896	0.909	0.915	0.911
	ZeroR	0.507	0.712	0.593	0.765	0.875	0.816

Table 6. Performance of WEKA Classifiers of classes misc and trees on Oper1 and CausFunc0 data sets, P stands for precision, R – for recall, F – for *f*-measure.

Classifier class	Classifier	Oper1			CausFunc0		
		P	R	F	P	R	F
misc	HyperPipes	0.785	0.769	0.775	0.865	0.838	0.849
	VFI	0.846	0.849	0.847	0.875	0.856	0.864
trees	ADTree	0.916	0.916	0.916	0.913	0.919	0.915
	BFTree	0.917	0.917	0.917	**0.921**	**0.925**	**0.922**
	DecisionStump	0.819	0.769	0.709	0.765	0.875	0.816
	FT	**0.92**	**0.92**	**0.92**	0.91	0.912	0.911
	Id3	**0.926**	**0.926**	**0.926**	0.905	0.905	0.905
	J48	0.91	0.91	0.91	0.909	0.916	0.91
	J48graft	0.907	0.907	0.907	0.897	0.907	0.895
	LADTree	**0.921**	**0.922**	**0.921**	0.919	0.919	0.919
	RandomForest	0.87	0.863	0.865	0.897	0.906	0.899
	RandomTree	0.819	0.813	0.816	0.865	0.876	0.87
	REPTree	0.903	0.903	0.903	**0.925**	**0.929**	**0.927**
	SimpleCart	**0.921**	**0.922**	**0.921**	**0.921**	**0.925**	**0.922**

Table 7 presents the highest state-of-the-art results [18] for detection of Oper1 and CausFunc0. NN signifies the Nearest Neighbor technique, NB – Naïve Bayesian network, ID3 – a decision tree classification technique based on the ID3-algorithm, TAN – Tree-Augmented Network Classification technique. Experiments on ID3-algorithm were not done for CausFunc0.

Table 7. State-of-the-art results [18] for Oper1 and CausFunc0, P is precision, R – recall, F – *f*-measure.

LF	NN		NB		ID3		TAN	
	P	R	P	R	P	R	P	R
Oper1	0.65	0.55	0.87	0.64	0.52	0.51	0.75	0.49
CausFunc0	0.59	0.79	0.44	0.89	--	--	0.45	0.57

The best state-of-the-art result (recall 0.89) is achieved for CausFunc0 by applying Naïve Bayesian network. The highest result we obtained (recall 0.933) is for CausFunc0 by applying rules.JRip classifier. (Names of classifiers are given in the format <className>.<classifierName>). In both cases, results for CausFunc0 are higher than for Oper1. It means that instances of Oper1 have more similarity with the rest of examples in the list of verb-noun pairs than instances of CausFunc0.

The training set for CausFunc0 in [18] contained 53 positive examples and for Oper1 – 84 positive examples. In our experiments, the training set for CausFunc0 had 102 positive examples, and for Oper1 – 157 positive examples. Larger data sets improve the performance of classifiers and the obtained results are more statistically reliable.

A difference between data representation in our experiments and data sets used in [18] should be noted here. In [18], every word in the training set was accompanied by its synonyms and hyperonyms, its own Base Concepts (BC) and the BCs of its hyperonyms, its own Top Concepts (TC) and the TCs of its hyperonyms taken from the Spanish part of the EuroWordNet. We included only hyperonyms in the training sets. Though WEKA classifiers were fed with less information in our case, it seems quite sufficient to produce better performance than in [18]. This phenomenon may remind us of the original intent of WordNet compilers who suggested to describe the meaning of any word by semantic relations only [11], like "is-a-kind-of" semantic relation of hyperonym hierarchy. Later, WordNet authors admitted that their previous assumption had been wrong and glosses were added to distinguish synonym sets. Though practical significance of glosses is generally accepted, we have seen that classifier accuracy is no worse if only hyperonyms are taken into account. Further research is needed to investigate how additional information, like that of semantic ontologies, changes classifier performance.

In the light of our results, let us consider the three methods of human recognition of collocations proposed in [18] and considered in Section 2.2.

Method 1. Collocations can be recognized by their similarity to the prototypical sample of each collocational type; this was modeled by Nearest Neighbor technique. Weka implements the nearest neighbor method in the following classifiers: rules.NNge, lazy.IB1, lazy.IBk and lazy.KStar [20]. Among these four classifiers, good results are obtained by rules.NNge for both Oper1 and CausFunc0. It demonstrates that Method 1 is feasible though does not produce very high quality results.

Method 2. Collocations can be recognized by similarity of semantic features of collocational elements to semantic features of elements of collocations known to belong to a specific LF; this was modeled by Naïve Bayesian network and a decision tree classification technique based on the ID3-algorithm. We tested three WEKA Naïve Bayesian classifiers – bayes.NaiveBayes, bayes.NaiveBayesSimple, bayes.NaiveBayesUpdateable [20]. All three classifiers show equal results, and the results for CausFunc0 are higher than for Oper1. ID3 algorithm is implemented in trees.Id3. This classifier gives better results than Naive Bayes and rules.NNge in Method 1.

Method 3. The third method was modeled by Tree-Augmented Network (TAN) Classification technique. As it is seen from Table 7, nearest neighbor algorithm gives better results in terms of recall than TAN. We did not apply TAN method in our experiments.

As it was mentioned before, the highest result obtained (recall 0.933) is produced by rules.JRip classifier. JRip classifier implements the RIPPER propositional rule learner [4]. The learning model is developed by iteration over a training subset, and by doing structure optimization to minimize error rate. More details can be found in [20].

5 Conclusions and future work

Our experiments have shown that verb-noun collocations can be classified according to semantic taxonomy of lexical functions using WEKA learning toolset. The best performance was demonstrated by rules.JRip classifier for lexical function CausFunc0. The highest result for detecting the lexical function Oper1 is given by bayes.BayesianLogisticRegression classifier. Both classifiers can be applied for high quality semantic annotation of verb-noun collocations based on the taxonomy of lexical functions. This was demonstrated on Spanish material.

As future work, we plan to experiment with different ratios of training and test sets as well as experiment on English verb-noun collocations. We will evaluate the performance of WEKA classifiers for more lexical functions and analyze errors of classifiers.

We have seen that rules.JRip accuracy is high when a verb-noun collocation is represented as a set of all hyperonyms of the noun and all hyperonyms of the verb. We plan to explore how performance of classifiers changes if we add other data like word glosses to the training set.

Acknowledgements. We are grateful to Adam Kilgarriff and Vojtech Kovár for providing us a list of most frequent verb-noun pairs from the Spanish Web Corpus of the Sketch Engine, www.sketchengine.co.uk.

The work was done under partial support of Mexican Government: SNI, COFAA-IPN, PIFI-IPN, CONACYT grant 50206-H, and SIP-IPN grant 20100773.

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. Apresjan, Ju.D., Boguslavsky, I. M., Iomdin, L. L., Tsinman, L. L.: Lexical Functions in NLP: Possible Uses. In: Klenner, M., Visser, H. (eds) Proceedings of the International Symposium held at the Ruprecht-Karls-Universität Heidelberg, 21-22 July 2000, pp. 55--72. Frankfurt am Main (2002)
3. Benson, M., Benson, E., Ilson, R.: The BBI Combinatory Dictionary of English: A Guide to Word Combinations. John Benjamin Publishing Company (1997)
4. Cohen, W. W.: Fast effective rule induction. In: Prieditis, A., Russell S. (eds) Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, pp.115—123. Morgan Kaufmann, San Francisco (1995)
5. Gitsaki, C.: The Development of ESL Collocational Knowledge, Ph.D. thesis, Center for Language Teaching and Research, The University of Queensland, Brisbane, Australia (1996)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten I. H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations, Volume 11, Issue 1 (2009)
7. Kilgarriff, A., Rychly, P., Smrz, P., Tugwell, D.: The Sketch Engine. In Proceedings of EURALEX 2004, pp. 105--116 (2004)
8. Mel'cuk, I. A.: Opyt teorii lingvisticeskix modelej "Smysl ? Tekst" ('A Theory of the Meaning-Text Type Linguistic Models'). Nauka, Moscow (1974)
9. Mel'cuk, I. A.: Phrasemes in Language and Phraseology in Linguistics. In: Everaert, M., Van der Linden, E.-J., Schenk, A., Schreuder, R. (eds) Idioms: Structural and Psychological Perspectives, pp. 167--232. Lawrence Erlbaum, Hillsdale, NJ (1995)

10.Mel'cuk, 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, pp. 37--102. Benjamins Academic Publishers, Amsterdam, Philadelphia, PA (1996)

11.Miller, G.A: Foreword. In: Fellbaum, C. (ed.) WordNet. An Electronic Lexical Database, pp. xv–xxii. MIT Press, Cambridge, Mass. (1998)

12.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)

13.Sinclair, J.: Corpus Concordance Collocation. OUP, Oxford (1991)

14.Spanish Web Corpus in SketchEngine, http://trac.sketchengine.co.uk/wiki/Corpora/SpanishWebCorpus

15.Spanish WordNet, http://www.lsi.upc.edu/~nlp/web/index.php?Itemid=57&id=31&option=com_content&task=view, last viewed March 26, 2010

16.Vossen P. (ed): EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Kluwer Academic Publishers, Dordrecht (1998)

17.Wanner, L.: Towards automatic fine-grained classification of verb-noun collocations. Natural Language Engineering, vol. 10(2), pp. 95--143. Cambridge University Press, Cambridge (2004)

18.Wanner, L., Bohnet, B., Giereth, M.: What is beyond Collocations? Insights from Machine Learning Experiments. EURALEX (2006)

19.WEKA Manual for Version 3-6-2, http://iweb.dl.sourceforge.net/project/weka/documentation/3.6.x/WekaManual-3-6-2.pdf

20.Witten, I. H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann, San Francisco (2005)

21.The University of Waikato Computer Science Department Machine Learning Group, WEKA download, http://www.cs.waikato.ac.nz/~ml/weka/index_downloading.html, last viewed March 26, 2010

22.The University of Waikato Computer Science Department Machine Learning Group, Attribute-Relation File Format, http://www.cs.waikato.ac.nz/~ml/weka/arff.html, last viewed March 26, 2010