

Supervised Learning for Semantic Classification of Spanish Collocations

Alexander Gelbukh and Olga Kolesnikova

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

Abstract. The meaning of word combination such as give a book or lend money can be obtained by mechanically combining the meaning of the two constituting words: to give is to hand over, a book is a pack of pages, then to give a book is to hand over a pack of pages. However, the meaning of such word combinations as give a lecture or lend support is not obtained in this way: to give a lecture is not to hand it over. Such word pairs are called collocations. While their meaning cannot be derived automatically from the meaning of their constituents, we show how to predict the meaning of a previously unseen word combination using semantic regularities we observe in a training set of collocations whose meaning has been specified manually.

1 Introduction

In natural language, individual words can either combine freely or form so-called collocations. The meaning of a free word combination can be understood from the meaning of its constituents. In particular, in automatic text understanding, semantic representation of a free word combination can be constructed by concatenating two definitions. For example, *to make* is ‘to create’, *a dress* is ‘a piece of clothes’, thus *to make a dress* is ‘to create a piece of clothes’. However, the combination *to make an announcement* means ‘to communicate information’ but not ‘to create information’. That is, simply concatenating the two definitions does not work here: while one word’s meaning is literal, the other’s is quite different. Such word combinations are called collocations. For example: *to make an announcement*, *to give a lecture*, *to pose an obstacle*.

The word used in a collocation in its literal, or typical, meaning is called the **base** of the collocation, and the other word, the **collocate**. In a collocation (e.g., *make an announcement*), the collocate (*make*) acquires a meaning (‘communicate’) different from its typical meaning (‘create’), i.e., the meaning it has in free word combinations (e.g., *make a dress*).

Collocations present a difficulty in automatic text understanding because the dictionaries usually do not include as senses all meanings that a word can have in collocations. One solution to this problem is to include all such meanings in the dictionary: with each word, add senses corresponding to all the meanings this word can have as a collocate in some collocation, along with the corresponding base. This, however, would be time and space consuming. What is more, such list of senses

would always be incomplete, since new collocations constantly appear in the language. A better solution is to build a system capable of predicting those new meanings on the fly.

In this paper we examine the latter approach. We look for semantic patterns in collocations and train a system to detect them, in order to predict a meaning of previously unseen collocations.

It has been observed in linguistic research that different collocates may express the same meaning. For example, consider collocations *deliver a lecture*, *take a walk*, *pose an obstacle*. Though the nouns functioning as bases in these collocations (*lecture*, *walk*, *obstacle*) show diverse semantics, the collocates expressed by the verbs have the same semantic content, namely ‘to do, perform, carry out something’. In collocations *plant a garden*, *give surprise*, *create a difficulty*, all the verbs have the meaning ‘to cause that something come into existence’. Thus, the first group of collocation has the semantic pattern “do what is denoted by the base” and the second group, “cause that what denoted by the base come into existence”. Semantic patterns can be specified using a formalism called **lexical function**¹.

Lexical function (LF) is defined in lexical semantics [2] as a function that associates a word with a corresponding word such that the latter expresses a given abstract meaning indicated by the name of lexical function. The concept of lexical function was introduced in linguistics as an element of functional models of natural language called *Meaning-Text Models*. The framework of the said models is *Meaning-Text Theory*. For details, see [6] and [3]. About 70 lexical functions have been identified. Each LF has been given a name in the form of an abbreviated Latin word whose meaning corresponds to the semantics of collocation. LF takes an input, termed the *keyword*, and produces the output, termed the *value* of LF. The notation of LF also includes information of syntactic structure of utterances where the keyword is used together with the LF’s value in a collocation. The syntactic structure is encoded with a string of integers put as an index after the LF’s name. Integer values specify semantic roles of LF’s keyword. For example, 1 stands for the agent, 2 stands for the patient, etc. Positions of integers in the string signify syntactic functions. First position in the string stands for subject, the second for direct object, etc. For example, the string *I2* says that the subject is the agent, and the direct object is the patient. We illustrate the above said with the examples given in Table 1, where K stands for *keyword*. Meanings of LFs are taken from [7].

Table 1 presents simple lexical functions which capture a single semantic element. There are collocations where the collocate may express a complex meaning including more than one semantic element. The semantic structure of such collocations are represented by complex semantic functions. Examples of some complex LFs are given in Table 3 of Section 4.

The rest of the paper is organized as follows. Section 2 defines the problem; Section 3 sketches related work and presents state of the art results. Section 4 describes the dataset and classification methods. The results obtained in our experiments are discussed in Section 5. Finally, Section 6 gives conclusions and speaks of future work.

¹ Not to be confused with a similar term used in computer programming.

2 Problem

Our task is to examine performance of supervised learning algorithms for classification of Spanish collocations according to the typology of LFs. The system is trained on a manually build corpus of verb-noun collocations annotated with LFs. Then the system is tested for recognition of eight LFs chosen for the experiments and the class of free word combinations (FWC), which gives us the total of nine semantic classes. We aim at detecting classifiers whose performance is best for each of these semantic classes.

Table 1. Examples of lexical functions

LF	Meaning	Keyword	Value	Syntactic structure	Example
Oper ₁	Lat. <i>operari</i> – ‘to do, carry out’	<i>support</i>	<i>lend</i>	Subject is the agent of K.	<i>The company lends support to charity.</i>
Oper ₂		<i>resistance</i>	<i>meet</i>	Subject is the patient of K.	<i>Allied Forces meet resistance in Afghanistan.</i>
Labor ₁₂	Lat. <i>laborare</i> – ‘to work, toil’	<i>control</i>	<i>keep under</i>	Subject is the agent of K, direct object is the patient of K.	<i>Sometimes people can not keep stress under control.</i>
Real ₁	Lat. <i>realis</i> – ‘real’. The values are verbs meaning ‘to fulfill the requirement of K’.	<i>obligation</i>	<i>fulfill</i>	The same as for Oper ₁ .	<i>Adult children must fulfill their obligation to care for elderly parents.</i>
Stop ₂	Lat. <i>stuppare</i> – ‘to stop up, to plug’. The values are verbs meaning ‘to stop functioning’.	<i>breath</i>	<i>loose</i>	The same as for Oper ₂ .	<i>She suddenly lost her breath and turned very pale.</i>

3 Related Work

It was mentioned in the Introduction that the concept of lexical functions was first elaborated in the frame of Meaning-Text Theory. Some research has been done on automatic detection of lexical functions. L. Wanner [13] proposed to view the task of LF detection as automatic classification of collocations according to LF typology. To fulfill this task, the nearest neighbor machine learning technique was used. Datasets included Spanish verb-noun pairs annotated with nine LFs. Every example in the datasets was represented by its hyperonyms retrieved from the Spanish part of EuroWordNet [12]. Every hyperonym was accompanied by its Basic Concepts and Top Concepts. A candidate instance was assigned that LF whose prototype was the most similar to the instance. Similarity was measured using path length in hyperonym hierarchy. These experiments gave the average F-measure of about 70%.

For classification of Spanish verb-noun collocations by LFs, Wanner *et al.* [14] applied four machine learning methods, namely, Nearest Neighbor technique, Naïve Bayesian network, Tree-Augmented Network Classification technique and a decision tree classification technique based on the ID3-algorithm. Experiments were carried out for two groups of verb-noun collocations. The first group included collocations where nouns belonged to the semantic field of emotions. In the second group, nouns were field-independent. In Section 5, we compare our results with those of [14] for verb-noun bigrams with field-independent nouns.

Alonso Ramos *et al.* [1] extracted collocations *support verb + object* from FrameNet corpus of examples [8]. Then they checked if the extracted collocations are of Oper_n . Their algorithm employ some syntactic, semantic and collocation annotations in the FrameNet corpus which serve as LF indicators. The proposed algorithm was tried out on a set of 208 instances and showed 76% accuracy. The authors come to a conclusion that it is feasible to extract and classify collocations according to LFs using semantically annotated corpora. Since the formalism of lexical function represents the correspondence between the keyword's semantic valency and of syntactic patterns together with semantic contents of collocation, such a conclusion sounds rather reasonable.

4 Data and Methodology

Our approach is based on supervised machine learning algorithms as implemented in the WEKA version 3-6-2 toolset [4, 15, 10]. Table 2 lists 68 classifiers tested for distinguishing between lexical functions. Table 3 presents LFs chosen for the experiments. LF meaning descriptions are taken from [7] and [13]; K stands for *keyword*. Examples are taken from the list of verb-noun pairs used in the experiments. The verb is the value of a corresponding LF, the noun is the keyword. Since examples are not grammatically well-formed utterances, the words are put in the form they appear in dictionary entries, with the exception of the verbs for Func_0 as explained further. The use of articles in the verb-noun pairs may vary depending on the context in which they are used in speech.

Now we give an explanation why the verbs for Func_0 are not used in their dictionary form in Table 3. This touches upon a peculiarity of the Spanish syntax. In Spanish, the pattern *verb + noun*, where the noun is the subject of the verb noun, is not as rare as in English, where the subject typically precedes the verb. For example, it is common to see such headings in newspapers as, for instance, *Sube inflación en Eurozona a 1.4%*, lit. *Rises inflation in Eurozone by 1.4%* (cited from the newspaper *Excelsior*, issue of April 16, 2010). Verbs which are values of Func_0 are used in utterances with their keywords as the subjects, so to give an example of a verb-noun pair where the verb is the value of Func_0 , we have to put the verb in the form that corresponds to the dictionary form of the noun, i.e., the verb has the form of 3rd person, singular number, present tense, indicative mood. The same is valid for the English translation. The words in the example *hace un mes* are always used in this word order. The corresponding English translation is *a month ago*. In English, there is no verb which is the exact equivalent of the verb *hacer* in the expression *hace un mes*, so *hacer* is translated not with a verb but the adverb *ago*.

For the experiments, we used the a list of the 1000 most frequent verb-noun pairs extracted automatically from the Spanish Web Corpus in the Sketch Engine [5]. All collocations in this list were annotated with lexical functions by human experts, who also tagged all the words in the collocations with word senses of the Spanish WordNet [9, 12]. The words in the pairs which are free word combinations (FWC) were also tagged with word senses and viewed as belonging to its own semantic class. Thus in our experiments, we treat FWC as a lexical function.

Table 2. WEKA classifiers trained to distinguish between lexical functions

AODE	ClassificationViaClustering	VFI
AODEsr	ClassificationViaRegression	ConjunctiveRule
BayesianLogisticRegression	CVParameterSelection	DecisionTable
BayesNet	Dagging	JRip
HNB	Decorate	NNge
NaiveBayes	END	OneR
NaiveBayesSimple	EnsembleSelection	PART
NaiveBayesUpdateable	FilteredClassifier	Prism
WAODE	Grading	Ridor
LibSVM	LogitBoost	ZeroR
Logistic	MultiBoostAB	ADTree
RBFNetwork	MultiClassClassifier	BFTree
SimpleLogistic	MultiScheme	DecisionStump
SMO	OrdinalClassClassifier	FT
VotedPerceptron	RacedIncrementalLogitBoost	Id3
Winnow	RandomCommittee	J48
IB1	RandomSubSpace	J48graft
IBk	RotationForest	LADTree
KStar	Stacking	RandomForest
LWL	StackingC	RandomTree
AdaBoostM1	ThresholdSelector	REPTree
AttributeSelectedClassifier	Vote	SimpleCart
Bagging	HyperPipes	

Input files were constructed in the Attribute-Relation File Format (ARFF) [11] accessible by WEKA classifiers. For each verb-noun pair, we used binary feature representation. For each word in the list of annotated verb-noun pairs, all its hyperonyms were retrieved from the Spanish WordNet referenced above, and the word itself is considered as the zero-level hyperonym. This gives 654 features to represent the nouns, and 280 features to represent the verbs. Each verb-noun pair in the training set is represented as a vector:

$$v_1, v_2, \dots, v_{654}, n_1, n_2, \dots, n_{280}, LF,$$

where v_n , n_k can be 0 or 1, and LF is a categorical feature having the value *yes* for positive instances of LF for which classification is done, and *no* for negative instances. Negative instances are collocations that belong to all other LFs in the list of collocations except to the LF chosen for classification.

The performance of WEKA classifiers was evaluated by comparing the values of precision, recall, and F-measure using 10-fold cross-validation. 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:

$$F = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1)$$

Table 3. Lexical functions chosen for the experiments

LF	Meaning	Collocation: LF value + keyword	
		Spanish	English translation
Oper ₁	Lat. <i>operare</i> – ‘to do, perform’. Experience (if K is an emotion), carry out K.	<i>alcanzar un objetivo</i> <i>aplicar una medida</i> <i>corregir un error</i> <i>satisfacer una necesidad</i>	<i>achieve a goal</i> <i>apply a measure</i> <i>correct a mistake</i> <i>satisfy a necessity</i>
Oper ₂	Undergo K, be source of K	<i>aprender una lección</i> <i>obtener una respuesta</i> <i>recibir ayuda</i> <i>sufrir un cambio</i>	<i>learn a lesson</i> <i>get an answer</i> <i>receive help</i> <i>suffer a change</i>
IncepOper ₁	Lat. <i>incipere</i> – ‘to begin’. Begin to do, perform, experience, carry out K.	<i>adoptar una actitud</i> <i>cobrar importancia</i> <i>iniciar una sesión</i> <i>tomar posición</i>	<i>take an attitude</i> <i>acquire importance</i> <i>start a session</i> <i>obtain a position</i>
ContOper ₁	Lat. <i>continuare</i> – ‘to continue’. Continue to do, perform, experience, carry out K.	<i>guardar silencio</i> <i>mantener el equilibrio</i> <i>seguir un modelo</i> <i>llevar una vida (ocupada)</i>	<i>keep silence</i> <i>keep one’s balance</i> <i>follow an example</i> <i>lead a (busy) life</i>
Func ₀	Lat. <i>functionare</i> – ‘to function’. K exists, takes place, occurs.	<i>el tiempo pasa</i> <i>hace un mes</i> <i>una posibilidad cabe la razón existe</i>	<i>time flies</i> <i>a month ago</i> <i>there is a possibility</i> <i>the reason exists</i>
CausFunc ₀	Lat. <i>causare</i> – ‘to cause’. Do something so that K begins occurring.	<i>encontrar respuesta</i> <i>establecer un sistema</i> <i>hacer campaña</i> <i>producir un efecto</i>	<i>find an answer</i> <i>establish a system</i> <i>conduct a campaign</i> <i>produce an effect</i>
CausFunc ₁	A person/object, different from the agent of K, does something so that K occurs and has effect on the agent of K.	<i>abrir camino</i> <i>causar daño</i> <i>dar respuesta</i> <i>producir un cambio</i>	<i>open the way</i> <i>cause damage</i> <i>give an answer</i> <i>produce a change</i>
Real ₁	Lat. <i>realis</i> – ‘real’. To fulfill the requirement of K, to act according to K.	<i>contestar una pregunta</i> <i>cumplir el requisito</i> <i>solucionar un problema</i> <i>utilizar la tecnología</i>	<i>answer a question</i> <i>fulfill the requirement</i> <i>solve a problem</i> <i>use technology</i>

5 Experimental Results

In Table 4, we present the main results obtained in our experiments. For each LF, we list three classifiers that have shown the best performance.

Table 4. Results showed by WEKA classifiers on the training set of lexical functions

LF	#	Classifier	P	R	F	Time	
						Train	Test
Oper1	157	BayesianLogisticRegression	0.879	0.866	0.873	1.22	0.50
		Id3	0.879	0.861	0.870	2.53	0.67
		SMO	0.862	0.866	0.864	5.19	0.39
Oper2	16	J48	0.923	0.571	0.706	0.72	0.45
		PART	0.923	0.571	0.706	0.72	0.45
		AttributeSelectedClassifier	0.923	0.571	0.706	1.58	0.42
IncepOper1	14	SMO	0.813	0.650	0.722	2.44	0.50
		NNge	0.923	0.600	0.727	1.61	1.17
		Prism	0.750	0.800	0.774	1.47	0.45
ContOper1	11	J48	0.833	0.769	0.800	0.30	0.44
		FilteredClassifier	0.833	0.769	0.800	0.38	0.47
		DecisionTable	0.909	0.769	0.833	10.0	0.30
Func0	22	AttributeSelectedClassifier	0.636	0.636	0.636	1.63	0.42
		HyperPipes	0.636	0.636	0.636	0.03	0.45
		BFTree	0.667	0.727	0.696	13.7	0.27
CausFunc0	102	REPTree	0.750	0.648	0.695	0.88	0.41
		EnsembleSelection	0.744	0.659	0.699	67.5	2.19
		JRip	0.747	0.705	0.725	0.97	0.39
CausFunc1	60	J48	0.842	0.696	0.762	1.22	0.42
		OrdinalClassClassifier	0.842	0.696	0.762	1.22	0.42
		END	0.842	0.696	0.762	1.39	0.42
Real ₁	45	Id3	0.600	0.574	0.587	2.33	0.66
		NNge	0.614	0.574	0.593	2.64	2.75
		FT	0.650	0.553	0.598	12.7	15.1
FWC	198	SMO	0.656	0.623	0.639	4.67	0.36
		BayesianLogisticRegression	0.658	0.629	0.643	0.89	0.45
		Prism	0.639	0.702	0.669	25.9	0.34

Total: 625

Average best: 0.737

Often, in classification experiments, the baseline is the performance of ZeroR classifier. ZeroR is a trivial algorithm that always predicts the majority class. It happens that the majority class in our training sets is always the class of negative instances. Even in the case of the LF which has the largest number of positive instances in the training set (198), the number of negative instances is still larger (427). Therefore, the ZeroR does not classify any test instances as positives, which gives always recall of 0 and undefined precision. Thus ZeroR is too bad a baseline to be considered.

In Table 4, the column marked by # specifies the number of positive instances for each LF. Recall that the whole dataset consists of all instances for all LFs mixed

together and contains 635 items, so the number of negative instances is 625 minus the value of the # column. For each classifier, the amount of time taken to build the model is given as well as the time taken to test the model on the training set. Note that these figures are meaningful because all LFs used the same set of 625 collocations as its training set and the same test sets in the tenfold cross-validation procedure.

As it is seen from Table 4, no single classifier is the best one for detecting all LFs. For each LF, the highest result is achieved by a different classifier. However, Prism reaches the highest F-score for both IncepOper₁ and FWC, though recall that FWC (free word combinations) is not a lexical function but is considered as an independent class along with LFs. The maximum F-measure of 0.873 is achieved by BayesianLogisticRegression classifier for Oper₁. The lowest best F-measure of 0.598 is shown by FT for Real₁. The average F-measure (calculated over only the nine best results, one for each LF) is 0.737.

The maximum time taken to build a model on the training data is shown by EnsembleSelection classifier for CausFunc₀, and the minimum time, by HyperPipes for Func₀. The maximum time taken to test the model is given by FT for Real₁, and the minimum time, by BFTree for Func₀.

We observed no correlation between the number of instances in the training set and the results obtained from the classifiers. For example, a low result is shown for the class FWC which has the largest number of positive examples. On the contrary, the second top result is achieved for LF ContOper₁, with the smallest number of positive examples. The minimum F-measure is obtained for Real₁ whose number of positive examples is about 77% smaller than the largest number of positive examples (FWC) and about 71% smaller than the number of positive examples for Oper₁, the detection of which was the best.

Table 5. State of the art results for some LFs taken from [14]

LF	NN			NB			ID3			TAN			Our
	P	R	F	P	R	F	P	R	F	P	R	F	F
Oper1	0.65	0.55	0.60	0.87	0.64	0.74	0.52	0.51	0.51	0.75	0.49	0.59	0.87
Oper2	0.62	0.71	0.66	0.55	0.21	0.30	N/A			0.55	0.56	0.55	0.71
ContOper1	N/A			N/A			0.84	0.57	0.70	N/A			0.83
CausFunc0	0.59	0.79	0.68	0.44	0.89	0.59	N/A			0.45	0.57	0.50	0.73
Real1	0.58	0.44	0.50	0.58	0.37	0.45	N/A			0.78	0.36	0.49	0.60

Average: 0.75

For comparison, Table 5 gives the state of the art results reported in [14] for LF classification using machine learning techniques. Out of nine LFs mentioned in [14] we give in Table 5 only those five that we used in our experiments, i.e., that are represented in Table 4. Also, as we have explained in Section 3, [14] reports the results for two different datasets: one for a narrow semantic field (that of emotions) and another for a field-independent (general) dataset. Since our dataset is also general, comparing them with a narrow-field dataset would not be fair, so in Table 5 we only give the field-independent figures from [14].

Not all methods have been applied in [14] for all LFs; if a method was not applied for a particular LF, the corresponding cells are marked as N/A. In this table, NN

stands for the Nearest Neighbor technique, NB for Naïve Bayesian network, ID3 is a decision tree classification technique based on the ID3-algorithm, and TAN for the Tree-Augmented Network Classification technique; P, R, and F are as in Table 4. In fact [14] did not give the value of recall, so we calculated it using (1). The last column repeats the best F-measure results from Table 4, for convenience of the reader. For each LF, the best result from [14], as well as the overall best result (including our experiments), are marked in boldface.

As seen from Table 5, for all LFs our experiments gave significantly higher figures than those reported in [14]. The best average F-measure from [14] is 0.66, while our experiments demonstrate the best average F-measure of 0.75. However, the comparison is not fair because different datasets have been used: the exact dataset used in [14] is unfortunately not available anymore;² ours is available from [16].

6 Conclusions and Future Work

We have shown that it is feasible to apply machine learning methods (specifically, those implemented in the WEKA toolkit) for predicting the meaning of unseen Spanish verb-noun collocations. Specifically, we trained a classifier to assign the semantic classes to a previously unseen collocation according to the formalism of lexical functions [7].

As features, both previous works and this work used the set of all hyperonyms of a word taken from WordNet [9, 12]. With this we re-confirmed that the set of hyperonyms can be used to describe lexical meaning and discriminate word senses.

Our experiments achieved the average F-measure of 74% (calculated basing on the best classifier for each of the nine LFs). This significantly outperforms the previously reported result of 66% [14] (our average result for the same subset of five LFs is 75%). However, the comparison is not fair because we used a dataset different from the one that has been used in [14], which is no longer available.

In the future, we plan to test other classification methods that were not examined in our experiments, both WEKA's modules that we did not yet test and methods not included in WEKA. We also plan to study the effect of other features, such as WordNet glosses. Finally, we intent to experiment with a word space models representing various similarity measures between collocations.

Acknowledgements. We are greatly thankful to Leo Wanner for useful and detailed discussions and for providing us lists of verb-noun collocations with partial manual markup. His kind help was very inspiring for us. We are also grateful to Adam Kilgarriff and Vojtěch Kovář 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. In: Proceedings of LREC, Marrakesh, Morocco, pp. 1154–1158 (2008)

² Personal communication with L. Wanner.

2. Apresjan, Ju. D.: Selected Works, Lexical Semantics, vol. 1. Vostochnaya Literatura Publishers, Moscow (1995) (in Russian)
3. Bolshakov, I.A., Gelbukh, A.F.: On Contemporary Status of the Meaning-Text Model. In: Guzman, A., Menchaka, R. (eds.) Selected Papers CIC-1999, CIC, IPN, Mexico City, pp. 17–25 (1999)
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
5. Kilgarriff, A., Rychly, P., Smrz, P., Tugwell, D.: The Sketch Engine. In: Proceedings of EURALEX 2004, pp. 105–116 (2004)
6. Mel'čuk, I.A.: A Theory of the Meaning-Text Type Linguistic Models. Nauka Publishers, Moscow (1974) (in Russian)
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, pp. 37–102. Benjamins Academic Publishers, Amsterdam (1996)
8. Ruppenhofer, J., Ellsworth, M., Petrucci, M., Johnson, C.R., Scheffczyk, J.: FrameNet II: Extended Theory and Practice. ICSI Berkeley (2006),
<http://framenet.icsi.berkeley.edu/book/book.pdf>
9. 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)
10. 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)
11. 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)
12. Vossen, P. (ed.): EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Kluwer Academic Publishers, Dordrecht (1998)
13. Wanner, L.: Towards automatic fine-grained classification of verb-noun collocations. Natural Language Engineering 10(2), 95–143 (2004)
14. Wanner, L., Bohnet, B., Giereth, M.: What is beyond Collocations? Insights from Machine Learning Experiments. In: EURALEX (2006)
15. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
16. <http://www.Gelbukh.com/lexical-functions>